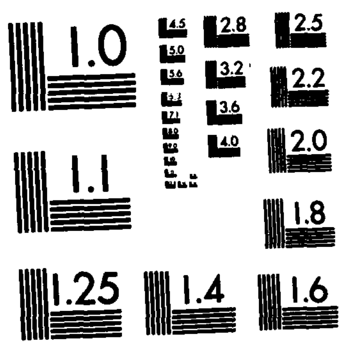
AD-A123 917 CALCULATION OF HYDROGRAPHIC POSITION DATA BY LEAST SQUARES ADJUSTMENT(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA F CASTRO E SILVA JUN 82 1/2

UNCLASSIFIED F/G 12/1 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6

(2)

AD A 123917

NAVAL POSTGRADUATE SCHOOL

Monterey, California

THESIS

CALCULATION OF HYDROGRAPHIC POSITION
DATA BY LEAST SQUARES ADJUSTMENT

by

Francisco Castro e Silva

June 1982

Thesis Advisor: Dudley Leath

Approved for public release; distribution unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A123917	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Calculation of Hydrographic Position Data by Least Squares Adjustment		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1982
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Francisco Castro e Silva		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		12. REPORT DATE June 1982
		13. NUMBER OF PAGES 190
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Least squares, Redundant observations, Fix by redundant observations

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

When redundant observations are available, hydrographic positioning problems require the application of a data adjustment method so that all information may be used for obtaining the most reliable "fix". One of the oldest and best engineering techniques developed for the purpose is based on the least squares principle. The theoretical background is provided to explain that principle and the technique for its application. Also, the

DD FORM 1473, 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

analytical solutions, and respective computer programs implementing them, are developed for the following hydrographic positioning methods: a) fix by N azimuths, b) fix by N sextant angles, c) fix by two range distances and one azimuth. For each method, an illustrative application of the respective computer program is presented.

The least squares adjustment method not only yields the most likely values for the fix coordinates but also statistically quantifies position accuracy. Relative accuracy achieved with conventional survey methods is elevated to absolute accuracy when redundant observations are made and adjusted using the method of least squares.

A

Approved for public release; distribution unlimited

Calculation of Hydrographic Position Data by Least Squares Adjustment

by

Francisco Castro e Silva
Lieutenant Commander, Portuguese Navy
Portuguese Naval Academy, 1967

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OCEANOGRAPHY (HYDROGRAPHY)

from the

NAVAL POSTGRADUATE SCHOOL
June 1982

Author: Francisco José Morgado Castro e Silva

Approved by: [illegible]
Thesis Advisor

Gerald Bryan Miller
Second Reader

[illegible]
Chairman, Department of Oceanography

[illegible]
Dean of Science and Engineering

ABSTRACT

When redundant observations are available, hydrographic positioning problems require the application of a data adjustment method so that all information may be used for obtaining the most reliable "fix". One of the oldest and best engineering techniques developed for the purpose is based on the least squares principle. The theoretical background is provided to explain that principle and the technique for its application. Also, the analytical solutions, and respective computer programs implementing them, are developed for the following hydrographic positioning methods:

a) fix by N azimuths;

b) fix by N sextant angles;

c) fix by two range distances and one azimuth.

For each method, an illustrative application of the respective computer program is presented.

The least squares adjustment method not only yields the most likely values for the fix coordinates but also statistically quantifies position accuracy. Relative accuracy achieved with conventional survey methods is elevated to absolute accuracy when redundant observations are made and adjusted using the method of least squares.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

ACKNOWLEDGMENT

For those that producing more than consuming, allowed me, during two years at least, to consume more than produce, my best thanks.

Monterey, the 6th of September 1981

LCDR Francisco Castro e Silva
Portuguese Navy

I. LEAST SQUARES ADJUSTMENT THEORY

In hydrographic surveying, the determination of position is as important as the measurement of depth. Conventional survey methods rely primarily on two lines of position (LOP) to establish a fix. These LOP's are obtained by measuring angles and distances directly. Alternately, electronic positioning systems are used to establish a pattern of LOP's (an electronic lattice) based on the propagation of electromagnetic energy.

Until recently it has been logistically unfeasable to obtain redundant observations in hydrographic surveying. However, with the advent of computers and miniaturized electronic positioning systems, redundant observations are now being made in order to increase fix accuracy and prevent delays due to equipment malfunction.

Mathematical adjustment methods must be applied to the redundant data sets in order to maximize the accuracy of each fix. One such adjustment method is based on the principle of least squares. It assumes that blunders and systematic errors have been removed from the data so that only random errors remain. This method yields not only the best estimate of position for a given set of redundant LOP's, but also assesses the absolute accuracy associated with each fix determination.

A. INTRODUCTION

In general, the redundant observations of any variables in a physical system (such as in hydrographic position determination) do not precisely satisfy the mathematical model developed to represent that system. However, the derivation of every mathematical model is based on the assumption that the true values of the variables will satisfy the model. The difference between the true and observed value for any physical variable is called the residual;

$$\text{residual} = \text{true value} - \text{observed value.} \quad (\text{I-1})$$

In making physical measurements, true values can never be determined. Considering the observed values as values assumed by random variables following normal distributions, every true value can be represented as the <u>mean</u> of a random variable. Therefore, eq. (I-1) can be rewritten as

$$\text{residual} = \text{mean of random variable} - \text{obs. value.} \quad (\text{I-2})$$

The least squares principle establishes a criterion for obtaining the best estimates of the true values. It states that the true values will be such that the sum of squared residuals is a minimum. For a further discussion of this principle, see Appendix A.

B. LEAST SQUARES PRINCIPLE FOR UNWEIGHTED OBSERVATIONS

Measuring different parameters of a mathematical or functional model, we associate with each parameter a random variable, X_i. Designating by γ_i the value assumed by the random variable (the observed value) and by μ_i its mean (the adjusted value), the residual V_i is given by

$$V_i = \mu_i - \gamma_i . \qquad (I-3)$$

For n observed parameters, the least squares fundamental condition is expressed as

$$\sum_{i=1}^{n} V_i^2 = V_1^2 + V_2^2 + \dots + V_n^2 = minimum$$

or, in matrix form

$$V^T V = minimum \qquad (I-4)$$

where $V^T = \left[V_1 \; V_2 \dots \; V_n \right]$.

C. LEAST SQUARES PRINCIPLE FOR WEIGHTED OBSERVATIONS

If the n observations are unequally weighted, then the least squares fundamental condition is expressed as

$$\sum_{i=1}^{n} \omega_i V_i^2 = \omega_1 V_1^2 + \dots + \omega_n V_n^2 = minimum$$

or, in matrix form

$$V^T W V = \text{minimum} \qquad (I-5)$$

where W is the $n \times n$ weight matrix. See Appendix B for a more complete discussion on the concept of weighted observations.

D. OBSERVATION EQUATIONS

In the expression for the residual, γ_i is a known value (the observed value) and μ_i represents (from a deterministic point of view) the true value, thus satisfying the relationship between the variables as expressed in the functional model. The model must define an analytical expression relating the unknown values with the known ones. In general, μ_i may be expressed as a function of the unknowns;

$$\mu_i = f_i \; (x_1 \; x_2 \; \cdots \; x_m)$$

where $x_1, x_2, \cdots, x_m$ are the unknowns. Therefore, eq. (I-3) can be rewritten as

$$V_i = f_i \; (x_1 \; x_2 \cdots \; x_m) - \gamma_i. \qquad (I-6)$$

The above expression is referred to as an observation equation.

If f_i is a linear function, the observation equation may be written as

$$V_i = a_{i0} + a_{i1} x_1 + \cdots + a_{im} x_m - \gamma_i \qquad (I-7)$$

where $a_{i0}, a_{i1}, \ldots, a_{im}$ are coefficients. The least squares method does not require that the observation equations be expressed in linear form. However, the computations to determine the values of the unknowns are greatly simplified if the observation equations are linearized.

If f_i is nonlinear, a Taylor's series expansion may be applied to linearize the function. Since it is not practical to work with all the terms of the expansion, only the zero and first order terms are used. Thus, the linearized form is an <u>approximate</u> analytical expression for f_i;

$$f_i = f_i\big|_0 + \left(\Delta x_1 \frac{\partial}{\partial x_1} + \cdots + \Delta x_m \frac{\partial}{\partial x_m}\right) f_i\big|_0 \,.$$

Since the function f_i and its partial derivatives may be evaluated given approximate "initial values" for the unknowns, the observation equations can be expressed as linear functions of the increments

$$f_i = a_{i0} + a_{i1} \Delta x_1 + \cdots + a_{im} \Delta x_m$$

where $a_{io} = f_i \big|_o$ and

$$a_{i1} = \frac{\partial f_i}{\partial x_1}\bigg|_o \quad , \ \ldots , \quad a_{im} = \frac{\partial f_i}{\partial x_m}\bigg|_o \ .$$

Therefore, the residual V_i will be stated as

$$V_i = a_{io} + a_{i1}\,\Delta x_1 + \ldots + a_{im}\,\Delta x_m - \gamma_i \ . \qquad (I\text{-}8)$$

It must be emphasized that, using the approximate expression for f_i, the least squares method will yield adjustments ($\Delta x_1, \cdots, \Delta x_m$) which must then be applied to the "initial approximations".

Therefore, an iterative solution is required to solve for the final values of the unknowns. The first adjusted results are used as the new initial values, and the observation equations must be formulated again. This process is continued until the increments become vanishingly small or, from a practical point of view, converge to within a specified tolerance.

E. LEAST SQUARES ADJUSTMENT METHOD

Considering eq. (I-7), and combining the constant terms, a new expression for the observation equation is obtained

$$V_i = a_{i1} x_1 + \cdots + a_{im} x_m - L_i \qquad (I-9)$$

where $$l_i = \gamma_i - a_{i0}.$$

Then, the n observation equations can be presented as the following system of n equations with m unknowns, where $n>m$ for the case of redundant observations:

$$\begin{cases} V_1 = a_{11} x_1 + \ldots + a_{1m} x_m - L_1 \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ V_n = a_{n1} x_1 + \ldots + a_{nm} x_m - L_n \end{cases} \qquad (I-10)$$

or, in matrix notation, as

$$V = AX - L \qquad (I-11)$$

where

$$V = \begin{bmatrix} V_1 \\ \vdots \\ V_n \end{bmatrix} \qquad A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \qquad L = \begin{bmatrix} l_1 \\ \vdots \\ l_n \end{bmatrix}.$$

Equations (I-10) and (I-11) express the general form of the observation equations. By imposing the least squares principle that $V^T W V =$ minimum, a set of equations are obtained which can be solved to find the best estimate of the unknown values. These expressions are known as the normal equations. For the observation equations as expressed in eq. (I-10), they form a set of m equations and m unknowns:

$$\begin{cases} [w_i a_{i1}^2] x_1 + \dots + [w_i a_{i1} a_{im}] x_m - [w_i a_{i1} l_i] = 0 \\ \dots\dots\dots\dots\dots\dots \\ [w_i a_{im} a_{i1}] x_1 + \dots + [w_i a_{im}^2] x_m - [w_i a_{im} l_i] = 0 \end{cases} \qquad (I\text{-}12)$$

where the brackets have the usual meaning of sum $(i = 1, 2, \dots, n)$.
In matrix notation, the normal equations are expressed as

$$(A^T W A) X = A^T W L. \qquad (I\text{-}13)$$

The normal equations are used to solve for the values of X;

$$X = (A^T W A)^{-1} (A^T W L) \qquad (I\text{-}14)$$

where X is the vector whose elements are the adjusted values for the unknowns. For a more complete development of the normal equations see Appendix C and Appendix D.

F. PRECISION OF OBSERVATIONS

When observing an unknown variable a finite number of times , n , the value of σ can be estimated by computing a sample standard deviation, S , according to the following formula:

$$S = \left[\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1} \right]^{1/2} = \left[\frac{\sum_{i=1}^{n} V_i^2}{n-1} \right]^{1/2}$$

where $x_i \ (i = 1, 2, \ldots, n)$ represents the observed values and $\bar{x}$ the average value, for a set of equally weighted observations.

Similarly, if m unknowns are (indirectly) observed n times, the best estimator for σ is the sample standard deviation, S, represented by the expression [REF. 1]

$$S = \left[\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-m} \right]^{1/2} = \left[\frac{\sum_{i=1}^{n} V_i^2}{r} \right]^{1/2}$$

assuming all observations are equally weighted. The value $r = n-m$ in the equation is known as the "degrees of freedom".

A set of n observations with assigned weights $\omega_1, \ldots, \omega_n$ is equivalent to a set of $\sum_{i=1}^{n} \omega_i$ observations which are all equally weighted. Thus, an artificial set of observed values is created in which ω_1 observations are equal to the 1st actual observed value, ω_2 observations are equal to the 2nd real observation, and so on for the remaining weighted observations.

Given an arbitrary set of weights, the set may be scaled so that the smallest weight has a value of ONE. This scale factor is known as the variance of unit weight, S_o^2. For a more complete discussion of this topic, see Appendix F.

Therefore, in the case of weighted observations, the best estimate for the value of the standard deviation of unit weight is given by

$$S_o = \left[\frac{\sum_{i=1}^{n} \omega_i v_i^2}{\left(\sum_{i=1}^{n} \omega_i \right) - m} \right]^{1/2} \qquad \text{(I-15)}$$

where m is the number of unknowns.

In matrix notation, eq. (I-15) is written as

$$S_o = \sqrt{\frac{V^T W V}{n - m}} \qquad \text{(I-16)}$$

where n , the number of unit weight observations, is given by the trace of weight matrix

$$n = \text{trace}(W) = \sum_{i=1}^{n} w_i . \qquad (I-17)$$

The standard deviation of the i^{th} observation (with weight w_i) is given by

$$S_i = \left[\frac{S_o^2}{w_i} \right]^{1/2} . \qquad (I-18)$$

G. PRECISION OF ADJUSTED VALUES

The elements of vector $X(x_1 \ldots x_m)$ given by

$$X = (A^T W A)^{-1} (A^T W L)$$

represent the adjusted values of the unknowns. The matrix $(A^T W A)^{-1}$ is known as the variance-covariance matrix Q, and individual elements are identified by the term q_{ij} .

The standard deviation of each adjusted value x_i is given by

$$S_{x_i} = S_o \sqrt{q_{ij}} \qquad (I-19)$$

where $j = i$, so that the q_{ij} terms are diagonal elements of the matrix $(A^T W A)^{-1}$.

The covariance between adjusted values x_i and x_j is given by

$$S_{x_i x_j} = S_o^2 q_{ij}. \qquad (\text{I-20})$$

For hydrographic position determination problems, the adjusted coordinates x and y correspond respectively to elements x_1 and x_2 of vector $X(x_1, x_2)$.

Therefore, the standard deviation of adjusted coordinates x and y is given as

$$\begin{cases} S_x = S_o \sqrt{q_{11}} \\ S_y = S_o \sqrt{q_{22}} \end{cases} . \qquad (\text{I-21})$$

The covariance between adjusted coordinates x and y is given by

$$S_{xy} = S_o^2 q_{12} \qquad (\text{I-22})$$

where factors q_{11}, q_{22} and q_{12} are elements of the symmetric square matrix

$$Q = (A^T W A)^{-1} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}.$$

For a more complete discussion of precision of adjusted values, see Appendix G.

H. THE ERROR ELLIPSE

Position errors are two dimensional and must be evaluated in terms of the errors along the x and y axes. Since the maximum and minimum errors do not necessarily occur along these axes, the orientation of these maximum and minimum errors must also be considered. Positioning errors may be evaluated in terms of the error ellipse (See Fig I-1).

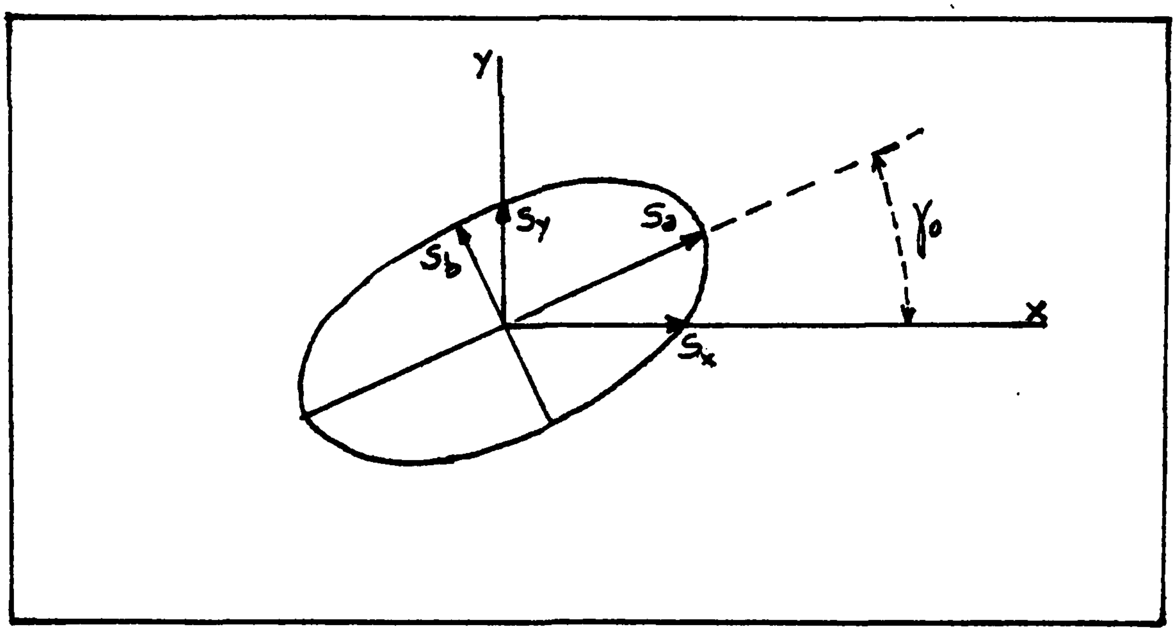


FIG I-1: ERROR ELLIPSE

The greater errors occur along a line making an angle γ_0 with the x-axis (measured anticlockwise) such that

$$\cot 2\gamma_0 = \frac{q_{11} - q_{22}}{2\, q_{12}} . \qquad (I-23)$$

The respective standard deviation is given by the semi-major axis length of the error ellipse

$$S_a = S_o \left[\frac{2\, q_{11}\, q_{22}}{q_{11} + q_{22} - D} \right]^{1/2} \qquad (I-24)$$

where

$$D = \left[(q_{11} - q_{22})^2 + 4\, q_{12}^2 \right]^{1/2} .$$

The smaller errors occur along a line perpendicular to S_a, and the respective standard deviation is given by the semi-minor axis length of the error ellipse

$$S_b = S_o \left[\frac{2\, q_{11}\, q_{22}}{q_{11} + q_{22} + D} \right]^{1/2} . \qquad (I-25)$$

The derivation of these equations is presented in Appendix H.

II. APPLICATION OF LEAST SQUARES ADJUSTMENT

The determination of a vessel's position at sea is a typical hydrographic problem for which the least squares adjustment is particularly well adapted. Various methods can be used for fix determinations. In this thesis, the following three methods will be presented:

a) fix by N azimuth angles

b) fix by N sextant angles

c) fix by two range distances and one azimuth angle.

For each method, the least squares adjustment is applied in the following manner:

1st: solution of the problem for particular conditions

2nd: numerical example

3rd: solution of the problem for general conditions

4th: formulation of an algorithm for the general conditions case

5th: implementation of the algorithm in Fortran language.

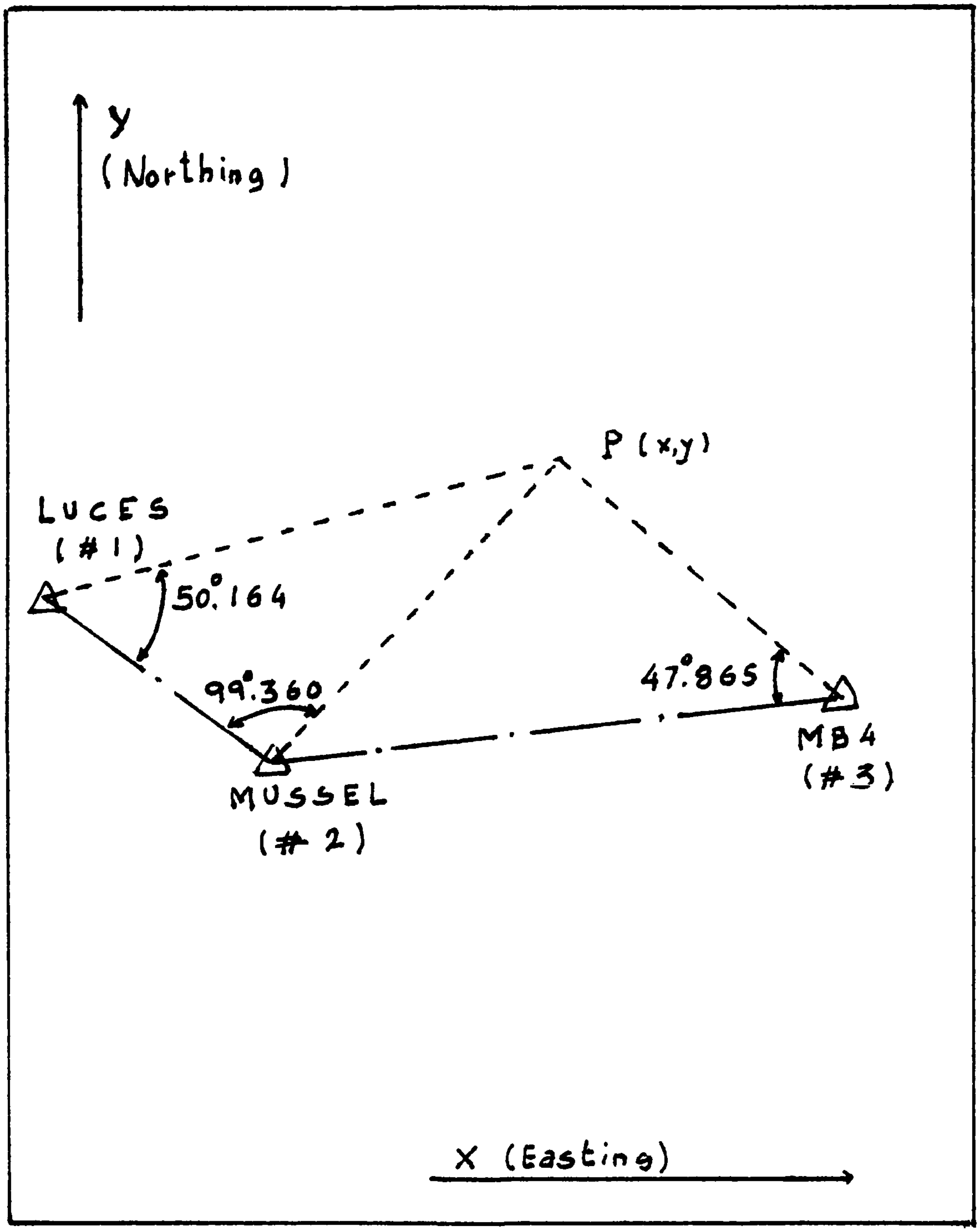


FIG II-1: FIX BY 3 AZIMUTHS

A. FIX DETERMINATION BY AZIMUTHS

1. Solution for Azimuths from 3 Stations

a. Determination of Adjusted Coordinates

Given a positioning problem as diagramed in FIG II-1, where:

A_{1P} - is the observed azimuth from station 1 to vessel position P

A_{2P} - is the observed azimuth from station 2 to vessel position P

A_{3P} - is the observed azimuth from station 3 to vessel position P

and

(x_1, y_1) - are the grid coordinates of station 1

(x_2, y_2) - are the grid coordinates of station 2

(x_3, y_3) - are the grid coordinates of station 3

the grid coordinates (x,y) of vessel position P will be determined.

Step 1) Formulation of observation equations

The analytical expression for the azimuth from station i (x_i, y_i) to P (x,y) is given by

$$Az_{iP} \text{ (radians)} = \tan^{-1} \frac{x - x_i}{y - y_i} = F(x,y).$$

The function $F(xy)$ must be expressed in a Taylor's series around an "initial position", P_0, whose coordinates

are defined as x_o and y_o . Evaluating the zero and first order terms of the series, the following expression is obtained:

$$Az_{iP} = \tan^{-1}\frac{x_o - x_i}{y_o - y_i} + \frac{(y_o - y_i)\,\Delta X - (x_o - x_i)\,\Delta Y}{(x_o - x_i)^2 + (y_o - y_i)^2} .$$

Designating the distance and azimuth from station $i\,(X_i, Y_i)$ to the "initial point" $P_o\,(x_o, y_o)$ by S_{io} and Az_{io} respectively, then

$$S_{io} = \left[(x_o - x_i)^2 + (y_o - y_i)^2 \right]^{1/2}$$

and

$$Az_{io} = \tan^{-1}\frac{x_o - x_i}{y_o - y_i} .$$

Therefore,

$$Az_{iP} = Az_{io} + \frac{y_o - y_i}{(S_{io})^2}\,\Delta X - \frac{x_o - x_i}{(S_{io})^2}\,\Delta Y ,$$

and the observation equations will be (for i=1,2,3)

$$V_i = \frac{y_o - y_i}{(S_{io})^2}\,\Delta X - \frac{x_o - x_i}{(S_{io})^2} - (A_{iP} - Az_{io})$$

where $A_{i\rho}$ is the observed azimuth. In the matrix form $AX - L = V$, the obs. equations will be

$$\begin{bmatrix} \dfrac{y_0 - y_1}{(S_{10})^2} & -\dfrac{x_0 - x_1}{(S_{10})^2} \\ \dfrac{y_0 - y_2}{(S_{20})^2} & -\dfrac{x_0 - x_2}{(S_{20})^2} \\ \dfrac{y_0 - y_3}{(S_{30})^2} & -\dfrac{x_0 - x_3}{(S_{30})^2} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} - \begin{bmatrix} A_{1\rho} - Az_{10} \\ A_{2\rho} - Az_{20} \\ A_{3\rho} - Az_{30} \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} .$$

The angles must be expressed in radians.

Step 2) Normal equations

Forming the normal equations, the adjusted values for Δx and Δy will be given by

$$X = (A^T W A)^{-1} \, (A^T W L)$$

where
$$X = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} .$$

Step 3) With the values Δx and Δy a new "initial point" $P_0'(x_0', y_0')$ will be obtained;

$$\begin{cases} x_0' = x_0 + \Delta x \\ y_0' = y_0 + \Delta y \end{cases}$$

and the procedure may be repeated in an iterative way until the increments Δx and Δy become vanishingly small. Then, the most probable values for the coordinates x and y will coincide with the coordinates of the last "initial point" obtained.

b. Precision of Observations and Adjusted Values

Step 1) From observation equations $AX-L=V$, where X has been obtained by the least squares method, the residuals are obtained, i.e., the differences between the "true" and observed values of the parameters.

Then, the standard deviation of unit weight is given by

$$S_0 = \left[\frac{\omega_1 V_1^2 + \omega_2 V_2^2 + \omega_3 V_3^2}{\omega_1 + \omega_2 + \omega_3 - m} \right]^{1/2}$$

where m is the number of unknowns observed. For that problem, the unknowns observed (indirectly) are x and y (m=2).

Therefore, in matrix notation, the above equation is expressed as

$$S_0 = \sqrt{\frac{V^T W V}{trace(W) - 2}}$$

where trace $(W) = \omega_1 + \omega_2 + \omega_3$.

Step 2) The standard deviation of each observation (with weight ω_i) is given by

$$S_i = \sqrt{\frac{S_0^2}{\omega_i}} \qquad (i = 1, 2, 3).$$

Step 3) The standard deviations of adjusted values are given by

$$S_x = S_o \sqrt{q_{11}}$$
$$S_y = S_o \sqrt{q_{22}} .$$

The covariance is given by

$$S_{xy} = q_{12} \cdot S_o^2 .$$

Step 4) The correlation coefficient ρ between x and y is given by

$$\rho = \frac{S_{xy}}{S_x S_y} = \frac{q_{12}}{\sqrt{q_{11} \cdot q_{22}}} .$$

c. Error Ellipse

Given the matrix

$$Q = (A^T W A)^{-1} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}$$

and the standard deviation S_o of a unit weight observation, the error ellipse parameters will be determined.

Step 1) Obtaining the value D,

$$D = \left[(q_{11} - q_{22})^2 + 4\, q_{12}^2 \right]^{1/2} .$$

Step 2) Semi-major axis,

$$S_a = S_o \left[\frac{2\, q_{11}\, q_{22}}{q_{11} + q_{22} - D} \right]^{1/2} .$$

Step 3) Semi-minor axis,

$$S_b = S_o \left[\frac{2\, q_{11}\, q_{22}}{q_{11} + q_{22} + D} \right]^{1/2} .$$

Step 4) Determining the angle γ_o (measured anticlockwise) from x-axis to semi-major axis:

4.1) The angle γ_o will satisfy

$$\tan(2\gamma_o) = \tan\Omega = \frac{2\, q_{12}}{q_{11} - q_{22}} .$$

4.2) For computer applications, Ω is defined to fall within the following limits: $-\pi/2 \leq \Omega \leq \pi/2$.

Then,

a) if $q_{11} = q_{22}$, choose $\gamma = \pi/4$

b) if $q_{11} \neq q_{22}$ and $\Omega \geq 0$, choose $\gamma = \Omega/2$

c) if $q_{11} \neq q_{22}$ and $\Omega < 0$, choose $\gamma = (\Omega + \pi)/2$

4.3) The intersection of error ellipse,

$$q_{22}\, x^2 - 2\, q_{12}\, xy + q_{11}\, y^2 - q_{11}\, q_{22}\, S_o^2 = 0$$

with the straight line $y = x \tan\gamma$, is given by x_1 and y_1,

such that

$$\begin{cases} x_1^2 = \dfrac{q_{11}\ q_{22}\ S_o^2}{q_{22} - 2\,q_{12}\tan\gamma + q_{11}\tan^2\gamma} \\ y_1^2 = x_1^2 \cdot \tan^2\gamma \end{cases}$$

4.4) Considering

$$D_1^2 = x_1^2 + y_1^2$$
$$D_o^2 = \left[(S_a + S_b)/2 \right]^2,$$

a) if $D_1^2 > D_o^2$, then $\gamma_o = \gamma$

b) if $D_1^2 < D_o^2$, then $\gamma_o = \gamma + \pi/2$.

2. Numerical Example

a. Determination of Adjusted Coordinates

The U.T.M. grid coordinates of shore stations, in FIG II-1, are:

COORDINATES	LUCES (#1)	MUSSEL (#2)	MB4 (#3)
x (EASTING)	595,794.5	597,967.8	603,425.2
y (NORTHING)	4,055,042.7	4,053,453.2	4,053,917.2

For illustrative purposes, the standard errors for azimuth observations made at each station were assigned the following values: $\sigma_1 = 0^\circ.02$, $\sigma_2 = 0^\circ.024$, $\sigma_3 = 0^\circ.018$.

The observed angles at each station were:

P - LUCES - MUSSEL = α_1 = $50.^\circ 164$,

P - MUSSEL - LUCES = α_2 = $99.^\circ 360$,

P - MB4 - MUSSEL = α_3 = $47.^\circ 865$.

Step 1) From the grid coordinates, the following azimuths between stations are obtained:

$$A_{12} = \tan^{-1}\left[(x_2 - x_1)/(y_2 - y_1)\right] = 126.^\circ 181,$$
$$A_{21} = A_{12} + 180^\circ = 306.^\circ 181,$$
$$A_{32} = \tan^{-1}\left[(x_2 - x_3)/(y_2 - y_3)\right] = 265.^\circ 140.$$

Step 2) The observed azimuths will be

$$A_{1P} = A_{12} - \alpha_1 = 76.^\circ 017,$$
$$A_{2P} = A_{21} + \alpha_2 = 45.^\circ 541,$$
$$A_{3P} = A_{32} + \alpha_3 = 313.^\circ 005.$$

Step 3) Formulation of observation equations

3.1) The first "initial point" is determined by the intersection of azimuth lines from stations #1 and #2 expressed by the equations

$$\begin{cases} y - y_1 = m_1 \, (x - x_1) \\ y - y_2 = m_2 \, (x - x_2). \end{cases}$$

Solving these equations simultaneously to find x and y, these values are used as the coordinates (x_o, y_o) of "initial

point", where

$$\begin{cases} X_0 = 600,877.5 \\ Y_0 = 4,056,308.4. \end{cases}$$

3.2) Determining azimuths between stations and "initial point" $P_0\ (X_0, Y_0)$,

$$Az_{10} = \tan^{-1}\left[(X_0 - X_1)/(Y_0 - Y_1)\right] = 76^\circ.017,$$

$$Az_{20} = \tan^{-1}\left[(X_0 - X_2)/(Y_0 - Y_2)\right] = 45^\circ.542,$$

$$Az_{30} = \tan^{-1}\left[(X_0 - X_3)/(Y_0 - Y_3)\right] = 313^\circ.185.$$

Then, evaluating the elements of the L matrix:

$$\begin{aligned} A_{1P} - Az_{10} &= 0^\circ.000 = 0.0 && \text{rad}, \\ A_{2P} - Az_{20} &= -0^\circ.001 = -0.000\ 0175 && \text{rad}, \\ A_{3P} - Az_{30} &= -0^\circ.180 = -0.003\ 1416 && \text{rad}. \end{aligned}$$

3.3) Determining squared distances between stations and "initial point" P_0,

$$\begin{aligned} (S_{10})^2 &= (X_0 - X_1)^2 + (Y_0 - Y_1)^2 = 27,438,885, \\ (S_{20})^2 &= (X_0 - X_2)^2 + (Y_0 - Y_2)^2 = 16,618,521, \\ (S_{30})^2 &= (X_0 - X_3)^2 + (Y_0 - Y_3)^2 = 12,208,613. \end{aligned}$$

3.4) Then, evaluating the elements of the A matrix,

$$(Y_0 - Y_1)/S_{10}^2 = 0.000\ 0461 \qquad -(X_0 - X_1)/S_{10}^2 = -0.000\ 1852$$

$$(Y_0 - Y_2)/S_{20}^2 = 0.000\ 1718 \qquad -(X_0 - X_2)/S_{20}^2 = -0.000\ 1751$$

$$(Y_0 - Y_3)/S_{30}^2 = 0.000\ 1959 \qquad -(X_0 - X_3)/S_{30}^2 = 0.000\ 2087.$$

3.5) Therefore, in matrix form, the observation equations are written as

$$\begin{bmatrix} 0.0000461 & -0.0001852 \\ 0.0001718 & -0.0001751 \\ 0.0001959 & 0.0002087 \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \end{bmatrix} - \begin{bmatrix} 0.000000 \\ -0.0000175 \\ -0.0031416 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}.$$

Step 4) Solution of normal equations

4.1) Determining the weight matrix,

$$\sigma_1 = 0.020 \longrightarrow 1/\sigma_1^2 = 2500$$
$$\sigma_2 = 0.024 \longrightarrow 1/\sigma_2^2 = 1736$$
$$\sigma_3 = 0.018 \longrightarrow 1/\sigma_3^2 = 3086.$$

Considering the least weight equal to ONE, it will be obtained that

$$\omega_1 = 1.44 \qquad \omega_2 = 1.00 \qquad \omega_3 = 1.78 \qquad \text{or} \qquad W = \begin{bmatrix} 1.44 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 1.78 \end{bmatrix}.$$

4.2) The solution of normal equations is given by

$$X = (A^T W A)^{-1} (A^T W L);$$

4.2.1) obtaining matrix

$$A^T W = \begin{bmatrix} 0.0000664 & 0.0001718 & 0.0003487 \\ -0.0002667 & -0.0001751 & 0.0003715 \end{bmatrix},$$

4.2.2) obtaining matrix

$$A^T W A = \begin{bmatrix} 0.000\,000\,10 & 0.000\,000\,03 \\ 0.000\,000\,03 & 0.000\,000\,20 \end{bmatrix},$$

4.2.3) obtaining matrix $Q = (A^T W A)^{-1}$

$$Q = \begin{bmatrix} 10,471,204 & -1,570,681 \\ -1,570,681 & 5,235,602 \end{bmatrix},$$

4.2.4) obtaining matrix

$$A^T W L = \begin{bmatrix} -0.000\ 001\ 1 \\ -0.000\ 001\ 2 \end{bmatrix},$$

4.2.5) finally, vector X is evaluated by solving the normal equations:

$$X = \begin{bmatrix} -9.6 \\ -4.6 \end{bmatrix}.$$

Step 5) First adjusted values of x and y

With the increments Δx and Δy a new "initial point" is obtained:

$$\begin{cases} x_0 = 600,877.5 - 9.6 = 600,867.9 \\ y_0 = 4,056,308.4 - 4.6 = 4,056,303.8 \end{cases}.$$

Step 6) With the new values, for the "initial point", the procedure indicated in steps 3.2, 3.3, 3.4, 3.5, 4.2 and 5 is repeated, and with the values now computed for ΔX and Δy a "closer" initial point is obtained.

Step 7) This procedure must be repeated, in an iterative way, until the increments Δx and Δy become vanishingly small, or, in practical terms, converging to within a specified tolerance. Then, the last "initial point" obtained will coincide with the most probable position for P(x,y).

b. Precision of Observations and Adjusted Values

Step 1) The residuals are obtained introducing $\Delta x = -9.6$ and $\Delta y = -4.6$ into the observation equations. Therefore,

$$V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \begin{bmatrix} 0.000\ 409\ 4 \\ -0.000\ 826\ 3 \\ 0.000\ 300\ 9 \end{bmatrix}.$$

Step 2) Obtaining scalar $V^T W V$,

$$V^T W V = 0.000\ 001\ 085 .$$

Therefore, $S_o = 0.000\ 6992$ radians .

Step 3) Obtaining standard deviation of each observation,

$$S_i = \sqrt{\frac{S_o^2}{\omega_i}} .$$

Then,

$$S_1 = 0.000\ 5827 \text{ rad} = 0^\circ.033$$
$$S_2 = 0.000\ 6992 \text{ rad} = 0^\circ.040$$
$$S_3 = 0.000\ 5241 \text{ rad} = 0^\circ.030 .$$

Step 4) Standard deviation and covariance of adjusted values x and y,

$$S_x = S_o\sqrt{q_{11}} = 2.26$$
$$S_y = S_o\sqrt{q_{22}} = 1.60$$
$$S_{xy} = S_o^2 q_{12} = -0.768 .$$

Note, S_x and S_y are expressed in the same units as the grid coordinates.

Step 5) Correlation coefficient,

$$\rho = \frac{S_{xy}}{S_x \cdot S_y} = -0.21 .$$

c. Error Ellipse

Given:

$$S_o = 0.000\,699\,2$$
$$q_{11} = 10,471,204.$$
$$q_{22} = 5,235,602 .$$
$$q_{12} = -1,570,681 .$$

the error ellipse parameters will be obtained.

Step 1) Determining D,

$$D = \left[(q_{11} - q_{22})^2 + 4q_{12}^2\right]^{1/2} = 6,105,709 .$$

Step 2) Semi-major axis,

$$S_a = S_o\left[\frac{2\,q_{11}\,q_{22}}{q_{11} + q_{22} - D}\right]^{1/2} = 2.36 .$$

Step 3) Semi-minor axis,

$$S_b = S_o \left[\frac{2 q_{11} q_{22}}{q_{11} + q_{22} + D} \right]^{\frac{1}{2}} = 1.57 .$$

Note, S_a and S_b are expressed in the same units as the grid coordinates.

Step 4) Determining the angle γ_o (measured anticlockwise) between x-axis and semi-major axis S_a

4.1) The solution of equation

$$\tan \Omega = \frac{2 q_{12}}{q_{11} - q_{22}}$$

is $\Omega = -30^\circ.964$.

4.2) Since $q_{11} \neq q_{22}$ and $\Omega < 0$, then

$$\gamma = \frac{\Omega + 180^\circ}{2} = 74^\circ.52 .$$

4.3) Obtaining X_1^2 and Y_1^2,

$$\begin{cases} X_1^2 = 0.175 \\ Y_1^2 = 2.28 . \end{cases}$$

4.4) Obtaining D_1^2 and D_o^2,

$$\begin{cases} D_1^2 = X_1^2 + Y_1^2 = 2.46 \\ D_o^2 = [(S_a + S_b)/2]^2 = 3.86 . \end{cases}$$

4.5) Since $D_1^2 < D_2^2$, then

$\gamma_o = \gamma + 90^\circ = 164^\circ.52$ (See FIG II-2).

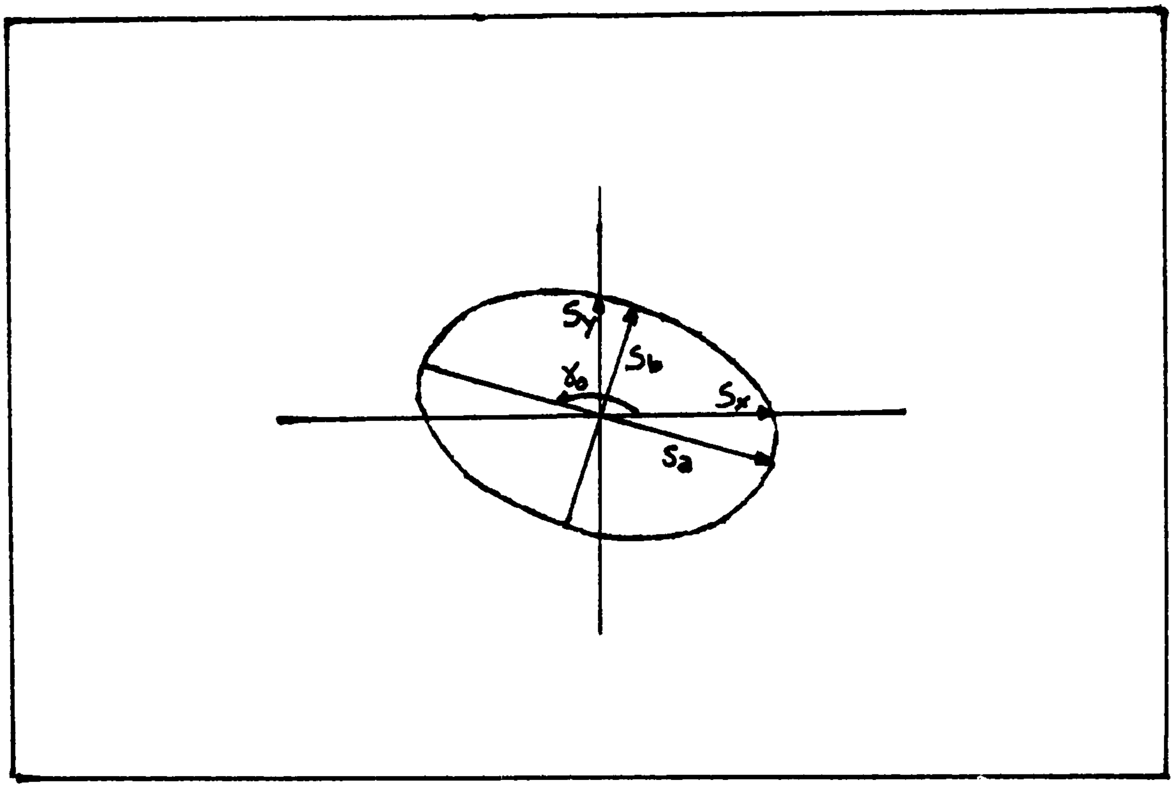


FIG II-2: ERROR ELLIPSE

All of these computations may be compared with those shown in the computer output section on page 148 . Differences in the results are due to the fact that the calculations illustrated on the preceeding pages were only carried out for one iteration.

3. Solution for the General Case

The solution will be presented in a way that can easily be implemented by an algorithm satisfying a modular design.

a. Determination of Adjusted Coordinates

Given:

a) the grid coordinates of N statons $S_i(x_i y_i)$, where the x-coordinate represents EASTING's and the y-coordinate represents NORTHING's

b) the azimuths A_{iP} $(i=1,\ldots,N)$ from stations S_i to vessel's position $P(x,y)$

c) and the standard deviations σ_i $(i=1,2,\ldots,N)$ of observed azimuths,

the adjusted coordinates for P(xy) will be determined.

Step 1) Weight matrix W

1.1) Squaring the inverse of standard deviations σ_i,

$$\omega'_i = \frac{1}{\sigma_i^2} \qquad (i=1,2,\cdots,N).$$

1.2) Designating by ω'_K the least ω'_i, the weights ω_i will be obtained;

$$\omega_i = \frac{\omega'_i}{\omega'_K} \qquad (i=1,2,\cdots,N).$$

1.3) The elements of square matrix W will be such that

$$\omega_{ij} = \begin{cases} 0, & \text{for } i \neq j \\ \omega_i, & \text{for } i = j \end{cases} \qquad (i,j=1,2,\ldots,N).$$

Step 2) Observation equations

2.1) Determination of first "initial point"

2.1.1) Designate by A_{KP} an observed azimuth A_{iP} $(i=2,3,\ldots,N)$ such that

$$\tan A_{KP} \neq \tan A_{1P}.$$

If no such azimuth is available, then the vessel's position is undetermined.

2.1.2) The intersection of the azimuth line A_{KP} from station K (x_K, y_K) with the azimuth line A_{1P} from station 1 (x_1, y_1) determines the first "initial point" P_0 (x_0, y_0). Therefore,

a) if $A_{1P} = n\pi$ (n=0,1) , then P_0 will be given by

$$\begin{cases} x_0 = x_1 \\ y_0 = y_K + \tan\left(\frac{5\pi}{2} - A_{KP}\right) \cdot (x_1 - x_K), \end{cases}$$

b) if $A_{KP} = n\pi$ $(n=0,1)$, then P_0 will be given by

$$\begin{cases} x_0 = x_K \\ y_0 = y_1 + \tan\left(\frac{5\pi}{2} - A_{1P}\right) \cdot (x_K - x_1), \end{cases}$$

c) otherwise, P_0 will be given by

$$\begin{cases} x_0 = \dfrac{y_K - y_1 + m_1 x_1 - m_K x_K}{m_1 - m_K} \\ y_0 = y_1 + m_1 (x_0 - x_1) \end{cases}$$

where

$$m_1 = \tan\left[(5\pi/2) - A_{1P}\right]$$

$$m_K = \tan\left[(5\pi/2) - A_{KP}\right].$$

2.2) Determination of azimuths Az_{io} between stations $S_i(x_i, y_i)$ and "initial point" P_o

2.2.1) Two angles, Az_{io} and $(Az_{io}+\pi)$ satisfy the equation

$$Az_{io} = \tan^{-1} \frac{x_o - x_i}{y_o - y_i} \qquad (i = 1, 2, \ldots, N).$$

Also, Az_{io} must be a positive angle between 0 and 2π. Since, in general, calculators give a solution between $(-\pi/2)$ and $(+\pi/2)$, a criterion will be established for selecting the valid solution.

2.2.2) Criterion :

a) if $y_o = y_i$ and $x_o > x_i$, then $Az_{io} = \pi/2$

b) if $y_o = y_i$ and $x_o < x_i$, then $Az_{io} = 3\pi/2$

c) if $x_o = x_i$ and $y_o > y_i$, then $Az_{io} = 0$

d) if $x_o = x_i$ and $y_o < y_i$, then $Az_{io} = \pi$

For $x_o \neq x_i$ and $y_o \neq y_i$, designate by α_{io} the solution, given by a calculator, of

$$\alpha_{io} = \tan^{-1} \frac{x_o - x_i}{y_o - y_i} \qquad (i = 1, 2, \ldots, N).$$

Therefore,

e) if $\alpha_{io} > 0$ and $x_o > x_i$, then $Az_{io} = \alpha_{io}$

f) if $\alpha_{io} < 0$ and $x_o > x_i$, then $Az_{io} = \alpha_{io} + \pi$

g) if $\alpha_{io} > 0$ and $x_o < x_i$, then $Az_{io} = \alpha_{io} + \pi$

h) if $\alpha_{io} < 0$ and $x_o < x_i$, then $Az_{io} = \alpha_{io} + 2\pi$

2.3) Determination of elements l_i of matrix L:

$$l_i = A_{iP} - Az_{io} \quad (i = 1, 2, \ldots, N).$$

2.4) Determination of squared distances between $S_i(x_i\, y_i)$ and $P_o(x_o\, y_o)$:

$$S_{io}^2 = (x_o - x_i)^2 + (y_o - y_i)^2 \quad (i = 1, 2, \cdots, N).$$

2.5) Determination of elements a_{ij} $(i = 1, \ldots, N;\ j = 1, 2)$ of matrix A :

$$a_i = \frac{y_o - y_i}{(S_{io})^2} \quad (i = 1, 2, \ldots, N)$$

$$a_{i2} = -\frac{x_o - x_i}{(S_{io})^2} \quad (i = 1, 2, \cdots, N).$$

Step 3) Normal equations

3.1) Determine matrix $A^T W$ (a matrix 2 x N).

3.2) Determine matrix $A^T W A$ (a matrix 2 x 2).

3.3) Determine matrix $(A^T W A)^{-1}$ (a matrix 2 x 2).

Since A^TWA is a symmetric matrix, then its inverse matrix will be $Q = (A^TWA)^{-1}$, also symmetric, such that

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} Q_{11} & Q_{12} \\ Q_{21} & Q_{22} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

It can be shown that

$$Q_{11} = -A_{22} \,/\, (A_{12}^2 - A_{11} \,.\, A_{22})$$
$$Q_{12} = Q_{21} = A_{12} \,/\, (A_{12}^2 - A_{11} \,.\, A_{22})$$
$$Q_{22} = -A_{11} \,/\, (A_{12}^2 - A_{11} \,.\, A_{22}) \,.$$

3.4) Determine matrix A^TWL (a matrix 2 x 1).

3.5) Finally, determine

$$X = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = (A^TWA)^{-1} \; (A^TW L).$$

Step 4) First adjusted values

With the values Δx and Δy the coordinates of the new "initial point" P_o' (x_o', y_o') are obtained:

$$\begin{cases} x_o' = x_o + \Delta x \\ y_o' = y_o + \Delta y \,. \end{cases}$$

Step 5) 2nd iteration

For obtaining a "closer" initial point repeat the steps 2.2, 2.3, 2.4, 2.5, 3, and 4.

Step 6) Next iterations

Repeat Step 5 until Δx and Δy become vanishingly small, or, in practical terms, converging to within a specified tolerance.

Then, the adjusted values for x and y will coincide with the coordinates of the last "initial point" obtained.

b. Precision of Observations

Given N (number of stations) and the matrices A, X, W and L determine:

Step 1) Matrix of residuals V (a matrix Nx1)

$$V = AX - L$$

Step 2) standard deviation S_o of the unit weight observation

2.1) Obtain $V^T W V$ (a scalar).

2.2) Obtain trace of weight matrix W;

$$trace(W) = \sum_{i=1}^{N} \omega_{ii}$$

where ω_{ii} is a diagonal element of weight matrix W.

2.3) Finally, S_o (in radians) will be given by

$$S_o = \sqrt{\frac{V^T W V}{trace(W) - 2}} .$$

Step 3) Standard deviation S_i of each observation (with weight ω_i),

$$S_i = \frac{S_o}{\sqrt{\omega_i}} \qquad (i = 1, 2, \ldots, N)$$

where S_i is expressed in radians.

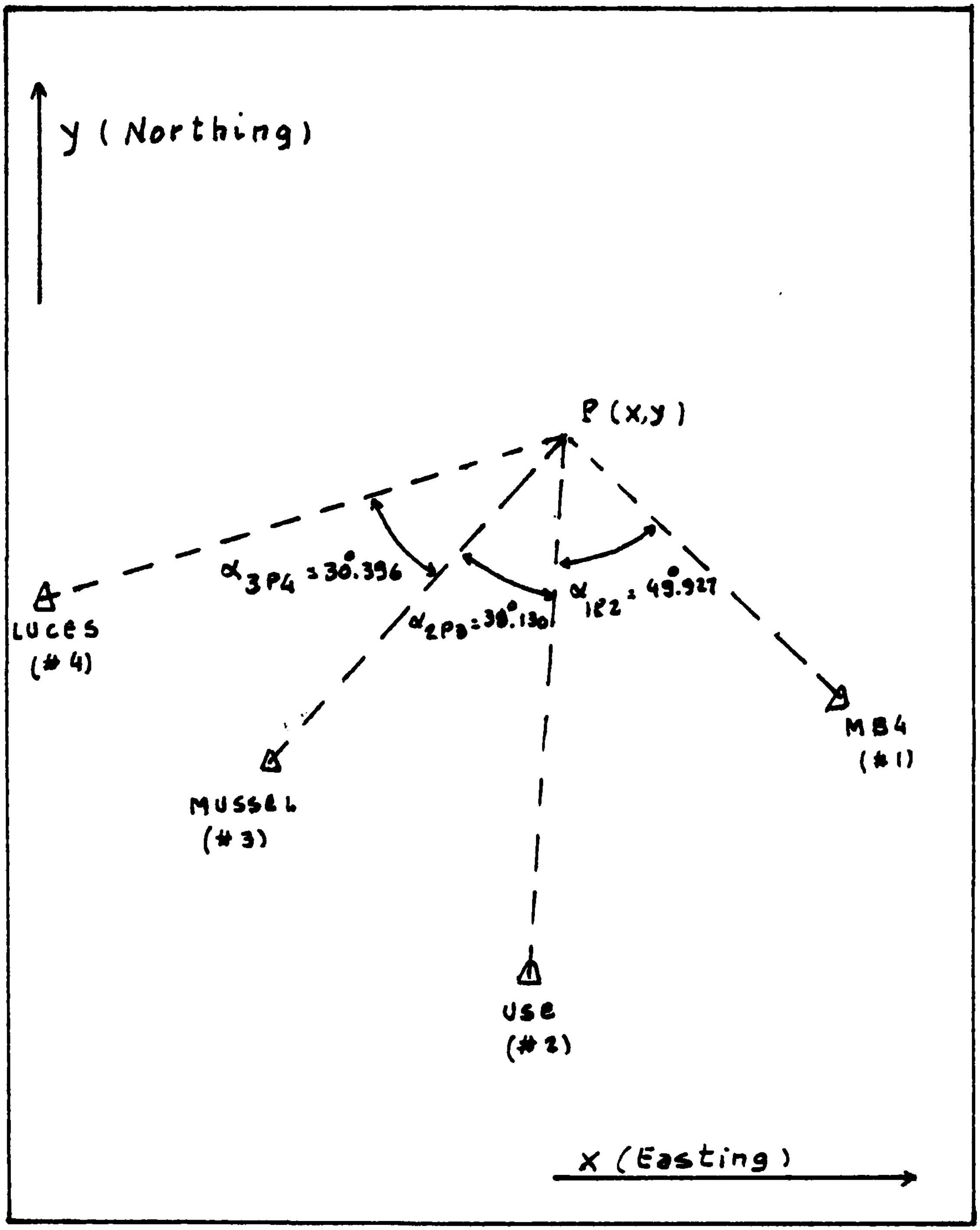


FIG II-3: FIX BY 3 SEXTANT ANGLES

B. FIX DETERMINATION BY SEXTANT ANGLES

1. Solution for 3 Sextant Angles (Between 4 Stations)

Given a positioning problem as diagramed in Fig II-3 in which:

α_{1P2} - is the observed sextant angle from P between stations 1 and 2

α_{2P3} - is the observed sextant angle from P between stations 2 and 3

α_{3P4} - is the observed sextant angle from P between stations 3 and 4

and

(x_1, y_1) - are grid coordinates of station 1

(x_2, y_2) - are grid coordinates of station 2

(x_3, y_3) - are grid coordinates of station 3

(x_4, y_4) - are grid coordinates of station 4

the grid coordinates (xy) of a vessel's position P will be determined.

Step 1) Formulation of observation equations

The analytical expression for the sextant angle $\alpha_{iP(i+1)}$, from the vessel's position P (x,y), between stations i (x_i, y_i) and $i+1$ (x_{i+1}, y_{i+1}) is given by

$$\alpha_{iP(i+1)} \text{ (in radians)} = Az_{P(i+1)} - Az_{Pi} =$$

$$\tan^{-1}\frac{x_{i+1} - x}{y_{i+1} - y} - \tan^{-1}\frac{x_i - x}{y_i - y} = F(x,y).$$

The function F(x,y) must be expressed in a Taylor's series around an "initial position", P_o , whose coordinates are defined as x_o and y_o . Evaluating the zero and first order terms of the series, the following expression is obtained:

$$F(xy) = Az_{p(i+1)} - Az_{pi} =$$

$$\tan^{-1}\frac{x_{i+1} - x_o}{y_{i+1} - y_o} - \tan^{-1}\frac{x_i - x_o}{y_i - y_o} +$$

$$\left[\frac{y_o - y_{i+1}}{(y_o - y_{i+1})^2 + (x_o - x_{i+1})^2} - \frac{y_o - y_i}{(y_o - y_i)^2 + (x_o - x_i)^2}\right]\Delta X +$$

$$\left[\frac{x_o - x_i}{(y_o - y_i)^2 + (x_o - x_i)^2} - \frac{x_o - x_{i+1}}{(y_o - y_{i+1})^2 + (x_o - x_{i+1})^2}\right]\Delta Y .$$

Designating by S_{oi} and $S_{o(i+1)}$, and by Az_{oi} and $Az_{o(i+1)}$, the distances and azimuths between "initial point" $P_o(x_o, y_o)$ and stations $i\,(x_i, y_i)$ and $i+1$ (x_{i+1}, y_{i+1}) , respectively, then

$$S_{oi} = \left[(x_i - x_o)^2 + (y_i - y_o)^2\right]^{1/2}$$

$$S_{o(i+1)} = \left[(x_{i+1} - x_o)^2 + (y_{i+1} - y_o)^2\right]^{1/2}$$

$$Az_{oi} = \tan^{-1}\left[(x_i - x_o) / (y_i - y_o)\right]$$

$$Az_{o(i+1)} = \tan^{-1}\left[(x_{i+1} - x_o) / (y_{i+1} - y_o)\right]$$

and,

$$\alpha_{i\ell(i+1)} = Az_{0(i+1)} - Az_{0i} + \left[(y_0 - y_{i+1})/(S_{0(i+1)})^2 - (y_0 - y_i)/(S_{0i})^2\right]\Delta x + \left[(x_0 - x_i)/(S_{0i})^2 - (x_0 - x_{i+1})/(S_{0(i+1)})^2\right]\Delta y .$$

Therefore, the observation equations may be expressed as (for $i = 1,2,3$)

$$V_i = \left[\frac{y_0 - y_{i+1}}{(S_{0(i+1)})^2} - \frac{y_0 - y_i}{(S_{0i})^2}\right]\Delta x + \left[\frac{x_0 - x_i}{(S_{0i})^2} - \frac{x_0 - x_{i+1}}{(S_{0(i+1)})^2}\right]\Delta y - \left[\alpha_{i\ell(i+1)} + Az_{0i} - Az_{0(i+1)}\right]$$

where $\alpha_{i\ell(i+1)}$ is the observed sextant angle. In matrix form, the above equation is expressed as

$$V = AX - L$$

where

$$A = \begin{bmatrix} \frac{y_0 - y_2}{(S_{02})^2} - \frac{y_0 - y_1}{(S_{01})^2} & \frac{x_0 - x_1}{(S_{01})^2} - \frac{x_0 - x_2}{(S_{02})^2} \\ \frac{y_0 - y_3}{(S_{03})^2} - \frac{y_0 - y_2}{(S_{02})^2} & \frac{x_0 - x_2}{(S_{02})^2} - \frac{x_0 - x_3}{(S_{03})^2} \\ \frac{y_0 - y_4}{(S_{04})^2} - \frac{y_0 - y_3}{(S_{03})^2} & \frac{x_0 - x_3}{(S_{03})^2} - \frac{x_0 - x_4}{(S_{04})^2} \end{bmatrix}$$

$$L = \begin{bmatrix} \alpha_{1P2} + Az_{01} - Az_{02} \\ \alpha_{2P3} + Az_{02} - Az_{03} \\ \alpha_{3P4} + Az_{03} - Az_{04} \end{bmatrix} \quad X = \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad V = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}.$$

In that result it should be noted that the angles are expressed in radians.

Step 2) Normal equations

Forming the normal equations, the adjusted values for Δx and Δy will be given by

$$X = (A^T W A)^{-1} (A^T W L).$$

Step 3) With the values Δx and Δy a new "initial point" $P'_o\ (x'_o, y'_o)$ is obtained,

$$\begin{cases} x'_o = x_o + \Delta x \\ y'_o = y_o + \Delta y, \end{cases}$$

and the procedure will be repeated in an iterative way until the increments Δx and Δy become vanishingly small.

Then, the most probable values for the coordinates x and y will coincide with the coordinates of the last "initial point" obtained.

2. Numerical Example

Referring to FIG II-3, the U.T.M. grid coordinates

of shore station are:

Coordinates	MB4 (#1)	USE (#2)	MUSSEL (#3)	LUCES (#4)
x(EASTING)	600,425.2	600,372.0	597,967.8	595,794.5
y(NORTHING)	4,053,917.2	4,051,216.9	4,053,453.2	4,055,042.7

The observed sextant angles are equally precise; thus, the weights will be

$$\omega_1 = \omega_2 = \omega_3 = 1,$$

and the weight matrix W is the identity matrix

$$W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The following sextant angles were measured:

MB4 - P - USE = α_{1P2} = $49^\circ.927$

USE - P - MUSSEL = α_{2P3} = $38^\circ.130$

MUSSEL - P - LUCES = α_{3P4} = $30^\circ.396$

Step 1) Formulation of observation equations

1.1) Determination of first "initial point" P_0 (x_0, y_0)

1.1.1) The first "initial point" P_0 will be the point determined by the two sextant angles α_{1P2} and α_{2P3} such that

$$\tan \alpha_{1P2} = \tan(Az_{02} - Az_{01}) = \frac{\dfrac{x_2 - x_0}{y_2 - y_0} - \dfrac{x_1 - x_0}{y_1 - y_0}}{1 + \dfrac{x_2 - x_0}{y_2 - y_0} \cdot \dfrac{x_1 - x_0}{y_1 - y_0}}$$

and

$$\tan \alpha_{2P3} = \tan(Az_{03} \quad Az_{02}) = \frac{\dfrac{x_3 - x_0}{y_3 - y_0} - \dfrac{x_2 - x_0}{y_2 - y_0}}{1 + \dfrac{x_3 - x_0}{y_3 - y_0} \cdot \dfrac{x_2 - x_0}{y_2 - y_0}} .$$

1.1.2) After some algebraic manipulation, it will be obtained that

$$y_0 = C\, x_0 + D$$

where

$$C = \frac{\dfrac{y_2 - y_1}{\tan \alpha_{1P2}} - \dfrac{y_3 - y_2}{\tan \alpha_{2P3}} + x_1 - x_3}{\dfrac{x_2 - x_3}{\tan \alpha_{2P3}} - \dfrac{x_1 - x_2}{\tan \alpha_{1P2}} - y_1 + y_3}$$

and

$$D = \frac{\dfrac{y_1 x_2 - y_2 x_1}{\tan \alpha_{1P2}} - \dfrac{y_2 x_3 - x_2 y_3}{\tan \alpha_{2P3}} + y_2 y_3 + x_2 x_3 - y_1 y_2 - x_1 x_2}{\dfrac{x_2 - x_3}{\tan \alpha_{2P3}} - \dfrac{x_1 - x_2}{\tan \alpha_{1P2}} - y_1 + y_3} .$$

1.1.3) The value x_0 will be a solution of equation

$$U x_0^2 + R x_0 + S = 0 \qquad (\text{II} - 1)$$

where $U = \tan \alpha_{1P2} \cdot (C^2+1)$

$$R = \tan \alpha_{1P2} \left[2CD - C(y_1+y_2) - (x_1+x_2) \right] - C(x_1-x_2) + y_1 - y_2$$

and

$$S = \tan \alpha_{1P2} \left[D^2 - D(y_1+y_2) + x_1 x_2 + y_1 y_2 \right] - D(x_1-x_2) - y_1 x_2 + y_2 x_1 \quad .$$

1.1.4) Two solution sets, (x_{01}, y_{01}) and (x_{02}, y_{02}), satisfy eq. (II-1). The valid solution corresponds to the solution set that, introduced into the following expression, yields the value that best approaches $\tan \alpha_{1P2}$:

$$\frac{(x_2-x_0)(y_1-y_0) - (x_1-x_0)(y_2-y_0)}{(y_2-y_0)(y_1-y_0) + (x_2-x_0)(x_1-x_0)} \rightarrow \tan \alpha_{1P2} \quad (\text{II}-2)$$

1.1.5) Using the numerical values

$\tan \alpha_{1P2} = 1.1887$ $\qquad$ $\tan \alpha_{2P3} = 0.7850$

$x_1 = 603{,}425.2$ $\qquad$ $y_1 = 4{,}053{,}917.2$

$x_2 = 600{,}372.0$ $\qquad$ $y_2 = 4{,}051{,}216.9$

$x_3 = 597{,}967.8$ $\qquad$ $y_3 = 4{,}053{,}453.2$

it will be obtained

$C = 11.109096$ $\qquad$ $D = -2{,}618{,}387.9$

$U = 147.885\ 403$ $\qquad$ $R = -1.776434 \times 10^8$

$S = 5.334\ 734\ 3 \times 10^{13}$.

The two solution sets satisfying eq (II-1) are

$$\begin{cases} x_{01} = 600,833 \\ y_{01} = 4,056,325 \end{cases} \quad \text{and}$$

$$\begin{cases} x_{02} = 600,390 \\ y_{02} = 4,051,405 \end{cases} .$$

Introducing the first solution set (x_{01}, y_{01}) into expression (II-2) the value 1.2923 will be obtained. Introducing (x_{02}, y_{02}) into the same expression, the value -0.9944 is obtained. Since the first set is the one that best approaches the value of $\tan \alpha_{1\ell 2} = 1.1887$, the coordinates of first "initial point" are

$$\begin{cases} x_0 = x_{01} = 600,833 \\ y_0 = y_{01} = 4,056,325 . \end{cases}$$

1.2) Determination of azimuths between "initial point" $P_0(x_0, y_0)$ and stations $S_i(x_i, y_i)$, (i=1,2,3,4) :

$$Az_{01} = \tan^{-1}\left[(x_1 - x_0)/(y_1 - y_0)\right] = 132^\circ.888$$

$$Az_{02} = \tan^{-1}\left[(x_2 - x_0)/(y_2 - y_0)\right] = 185^\circ.157$$

$$Az_{03} = \tan^{-1}\left[(x_3 - x_0)/(y_3 - y_0)\right] = 224^\circ.934$$

$$Az_{04} = \tan^{-1}\left[(x_4 - x_0)/(y_4 - y_0)\right] = 255^\circ.721 .$$

Then,

$$\alpha_{1P2} + Az_{01} - Az_{02} = -2.^\circ 342 = -0.040\,875\,6 \text{ rad}$$
$$\alpha_{2P3} + Az_{02} - Az_{03} = -1.^\circ 647 = -0.028\,745\,6 \text{ rad}$$
$$\alpha_{3P4} + Az_{03} - Az_{04} = -0.^\circ 391 = -0.006\,824\,2 \text{ rad}.$$

1.3) Determination of squared distances between P_0 and stations:

$$(S_{01})^2 = (x_1 - x_0)^2 + (y_1 - y_0)^2 = 12{,}517{,}002$$
$$(S_{02})^2 = (x_2 - x_0)^2 + (y_2 - y_0)^2 = 26{,}305{,}207$$
$$(S_{03})^2 = (x_3 - x_0)^2 + (y_3 - y_0)^2 = 16{,}456{,}606$$
$$(S_{04})^2 = (x_4 - x_0)^2 + (y_4 - y_0)^2 = 27{,}030{,}776 .$$

1.4) then,

$$\frac{y_0 - y_2}{(S_{02})^2} - \frac{y_0 - y_1}{(S_{01})^2} = 0.000\,001\,8 \qquad \frac{x_0 - x_1}{(S_{01})^2} - \frac{x_0 - x_2}{(S_{02})^2} = -0.000\,224\,6$$

$$\frac{y_0 - y_3}{(S_{03})^2} - \frac{y_0 - y_2}{(S_{02})^2} = -0.000\,019\,7 \qquad \frac{x_0 - x_2}{(S_{02})^2} - \frac{x_0 - x_3}{(S_{03})^2} = -0.000\,156\,6$$

$$\frac{y_0 - y_4}{(S_{04})^2} - \frac{y_0 - y_3}{(S_{03})^2} = -0.000\,127\,1 \qquad \frac{x_0 - x_3}{(S_{03})^2} - \frac{x_0 - x_4}{(S_{04})^2} = -0.000\,012\,3 .$$

1.5) Therefore, in matrix form, the observation equations will be expressed as

$$\begin{bmatrix} +0.000\,001\,8 & -0.000\,224\,6 \\ -0.000\,019\,7 & -0.000\,156\,6 \\ -0.000\,127\,1 & -0.000\,012\,3 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} - \begin{bmatrix} -0.040\,875\,6 \\ -0.028\,745\,6 \\ -0.006\,824\,2 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} .$$

Step 2) Normal equations

The solution of normal equations is given by

$$X = (A^T W A)^{-1} (A^T W L) ;$$

2.1) obtaining matrix $A^T W$,

$$A^T W = \begin{bmatrix} 0.000\ 001\ 8 & -0.000\ 019\ 7 & -0.000\ 127\ 1 \\ -0.000\ 224\ 6 & -0.000\ 156\ 6 & -0.000\ 012\ 3 \end{bmatrix}.$$

2.2) obtaining matrix $A^T W A$,

$$A^T W A = \begin{bmatrix} 1.65 \times 10^{-8} & 4.24 \times 10^{-9} \\ 4.24 \times 10^{-9} & 7.51 \times 10^{-8} \end{bmatrix}.$$

2.3) obtaining matrix $Q = (A^T W A)^{-1}$,

$$Q = \begin{bmatrix} 61,498,278 & -3,472,073 \\ -3,472,073 & 13,511,606 \end{bmatrix}.$$

2.4) obtaining matrix $A^T W L$,

$$A^T W L = \begin{bmatrix} 1.360 \times 10^{-6} \\ 1.377 \times 10^{-5} \end{bmatrix},$$

2.5) finally, vector X it will be obtained

$$X = \begin{bmatrix} 35.8 \\ 181.3 \end{bmatrix}.$$

Step 3) First adjusted values of x and y

With the increments Δx and Δy a new "initial point" is obtained;

$$\begin{cases} x_0 = 600,833 + 35.8 = 600,868.8 \\ y_0 = 4,056,325 + 181.3 = 4,056,506.3 . \end{cases}$$

Step 4) With the new values for the "initial point", the procedure indicated in steps 1.2, 1.3, 1.4, 1.5, 2 and 3 is repeated, and with the values now obtained for Δx and Δy a "closer" initial point is obtained.

Step 5) That procedure must be repeated, in an iterative way, until the increments Δx and Δy become vanishingly small, or, in practical terms, converging to within a specified tolerance. Then, the last "initial point" obtained will coincide with the most probable position for P(xy).

These computations may be compared with those shown in the computer output section on page 149 . Differences in the results are due to the fact that the calculations illustrated on the preceeding pages were only carried out for one iteration.

3. Solution for the General Case

The solution will be presented in such a way that easily can be implemented by an algorithm satisfying a modular design.

Given:

a) the grid coordinates of M=N+1 stations $S_i\ (x_i, y_i)$, ordered in a clockwise sense around vessel's position,

b) the N sextant angles $\alpha_{iP(i+1)}$ between stations $S_i\ (x_i, y_i)$ and $S_{i+1}\ (x_{i+1}, y_{i+1})$,

c) and the standard deviations σ_i (i=1,2,...,N) of observed sextant angles,

the adjusted coordinates for P(x,y) will be determined.

Step 1) Weight matrix W

Obtained as indicated on Step 1 of subsection II.A.3.a.

Step 2) Observed equations

2.1) Determination of first "initial point" P_o (x_o, y_o)

2.1.1) The first "initial point" $P_o(x_o, y_o)$ will be the point determined by the two sextant angles α_{1P2} and α_{2P3} such that

$$\tan\alpha_{1P2} = \tan(Az_{02} - Az_{01}) = \frac{\dfrac{x_2 - x_o}{y_2 - y_o} - \dfrac{x_1 - x_o}{y_1 - y_o}}{1 + \dfrac{x_2 - x_o}{y_2 - y_o} \cdot \dfrac{x_1 - x_o}{y_1 - y_o}}$$

and

$$\tan\alpha_{2P3} = \tan(Az_{03} - Az_{02}) = \frac{\dfrac{x_3 - x_o}{y_3 - y_o} - \dfrac{x_2 - x_o}{y_2 - y_o}}{1 + \dfrac{x_3 - x_o}{y_3 - y_o} \cdot \dfrac{x_2 - x_o}{y_2 - y_o}} .$$

2.1.2) Test for undetermined initial position (See FIG II-4).

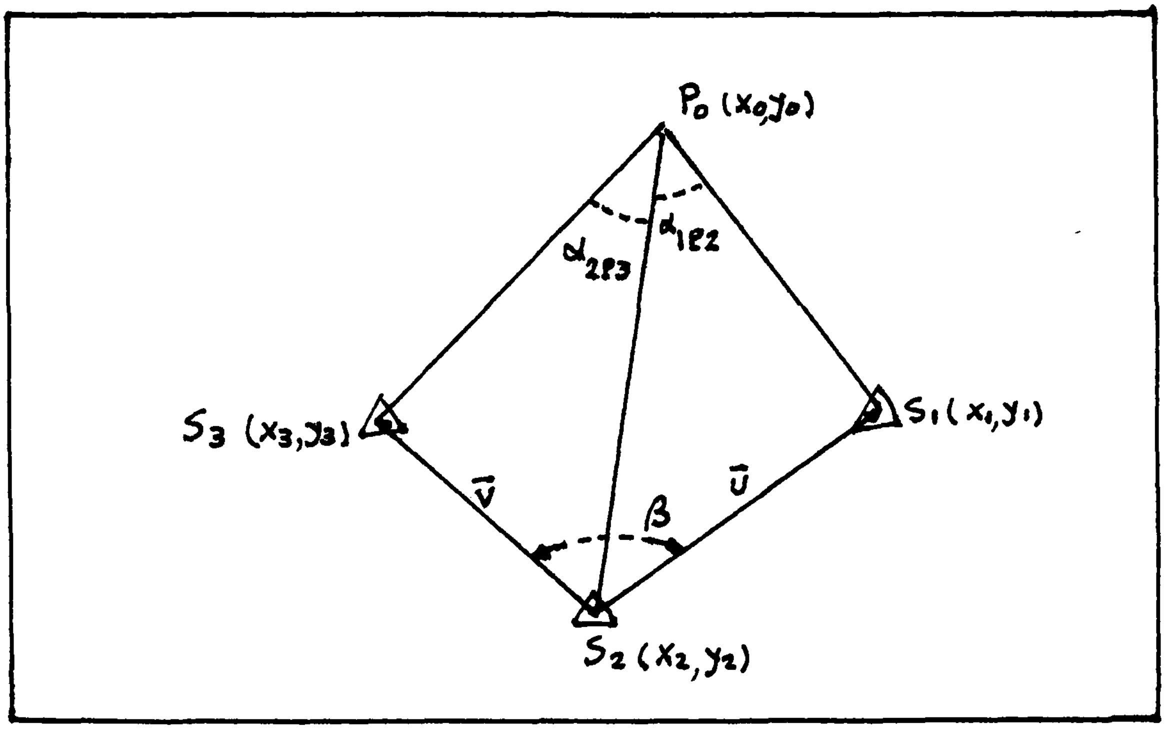


FIG II-4: UNDETERMINED FIX BY 2 SEXTANT ANGLES

If P_0, S_1, S_2 and S_3 belong to the same circumference, then

$$\beta = 180^\circ - (\alpha_{1P2} + \alpha_{2P3}).$$

Therefore, the dot product of vectors $\bar{U}$ and $\bar{V}$ (FIG II-4) will be

$$\bar{U} \cdot \bar{V} = (x_1 - x_2)(x_3 - x_2) + (y_1 - y_2)(y_3 - y_2) = |\bar{U}| \cdot |\bar{V}| \cos\beta .$$

So, the condition for an undetermined fix by two sextant angles will be

$$\cos(\alpha_{1P2} + \alpha_{2P3}) = \frac{(x_1 - x_2)(x_2 - x_3) + (y_1 - y_2)(y_2 - y_3)}{\left\{\left[(x_1 - x_2)^2 + (y_1 - y_2)^2\right]\left[(x_3 - x_2)^2 + (y_3 - y_2)^2\right]\right\}^{1/2}} .$$

2.1.3) Of the initial position is not undetermined, then its coordinates can be obtained as follows:

2.1.3.1) 1st case: $\alpha_{1P2} \neq 90^\circ$ and $\alpha_{2P3} \neq 90^\circ$

Let

$$A = \tan \alpha_{1P2}$$

$$B = \tan \alpha_{2P3}$$

$$E = \left[(x_2 - x_1)/A\right] + \left[(x_2 - x_3)/B\right] + y_3 - y_1$$

$$F = \left[(y_2 - y_1)/A\right] + \left[(y_2 - y_3)/B\right] + x_1 - x_3$$

$$G = (y_1 x_2 - y_2 x_1)/A + (x_2 y_3 - x_3 y_2)/B + x_2 x_3 + y_2 y_3 - x_1 x_2 - y_1 y_2 .$$

a) If $F=0$, then

$$y_0 = G/E = D .$$

Let

$$U = A$$

$$R = -A(x_1 + x_2) + y_1 - y_2$$

$$S = A\left[D^2 - D(y_1 + y_2) + x_1 x_2 + y_1 y_2\right] + D(x_2 - x_1) + x_1 y_2 - x_2 y_1 .$$

Then, from

$$U x_0^2 + R x_0 + S = 0 \qquad \text{obtain}$$

$$x_0 = \frac{-R \pm \sqrt{R^2 - 4US}}{2U} .$$

From solution sets (x_{o1}, y_o) and (x_{o2}, y_o), choose the one that best satisfies

$$\tan \alpha_{1\ell 2} = \frac{(x_2 - x_o)(y_1 - y_o) - (x_1 - x_o)(y_2 - y_o)}{(y_2 - y_o)(y_1 - y_o) + (x_2 - x_o)(x_1 - x_o)} . \qquad (\mathrm{II}-3)$$

b) If E=0, then

$$x_o = - G/F = H .$$

Let

$$U = A$$

$$R = - A(y_1 + y_2) - x_1 + x_2$$

$$S = A\left[H^2 - H(x_1 + x_2) + x_1 x_2 + y_1 y_2\right] + H(y_1 - y_2) + x_1 y_2 - x_2 y_1 .$$

Then, from

$U y_o^2 + R y_o + S = 0$ obtain

$$y_o = \frac{-R \pm \sqrt{R^2 - 4US}}{2U} .$$

From solution sets (x_0, y_{01}) and (x_0, y_{02}) choose the one that best satisfies equation (II-3).

c) If $E \neq 0$ and $F \neq 0$, then

$$y_0 = (F/E)\,x_0 + G/E = Cx_0 + D.$$

Let

$$U = A\,(C^2+1)$$

$$R = A\left[2CD - C(y_1+y_2) - (x_1+x_2)\right] - Cx_1 + Cx_2 - y_2 + y_1$$

$$S = A\left[D^2 - D(y_1+y_2) + x_1x_2 + y_1y_2\right] + D(x_2-x_1) + x_1y_2 - x_2y_1 .$$

Then, from

$$Ux_0^2 + Rx_0 + S = 0 \qquad \text{obtain}$$

$$x_0 = \frac{-R \pm \sqrt{R^2 - 4US}}{2U} .$$

From solution sets (x_{01}, y_{01}) and (x_{02}, y_{02}) choose the one that best satisfies equation (II-3).

2.1.3.2) 2nd case: $\alpha_{1P2} = 90^\circ$ and $\alpha_{2P3} \neq 90^\circ$

Let

$$B = \tan \alpha_{2P3}$$
$$E = B(y_3 - y_1) + x_2 - x_3$$
$$F = B(x_1 - x_3) + y_2 - y_3$$
$$G = B(y_2 y_3 + x_2 x_3 - y_1 y_2 - x_1 x_2) + x_2 y_3 - x_3 y_2 \,.$$

a) If $F = 0$, then

$$y_0 = G/E = D.$$

Let

$$R = -(x_1 + x_2)$$

$$S = D^2 - D(y_1 + y_2) + y_1 y_2 + x_1 x_2 \,.$$

Then, from

$$x_0^2 + R x_0 + S = 0$$ obtain

$$x_0 = \frac{-R \pm \sqrt{R^2 - 4S}}{2} \,.$$

From solution sets (x_{01}, y_0) and (x_{02}, y_0) choose the one that best satisfies

$$\tan \alpha_{2P3} = \frac{(x_3 - x_0)(y_2 - y_0) - (x_2 - x_0)(y_3 - y_0)}{(y_3 - y_0)(y_2 - y_0) + (x_3 - x_0)(x_2 - x_0)} \,. \qquad (\text{II}-4)$$

b) If $E = 0$, then

$$x_0 = -G/F = H \,.$$

Let

$$R = -(y_1 + y_2)$$
$$S = H^2 - H(x_1 + x_2) + x_1 x_2 + y_1 y_2 .$$

Then, from

$$y_o^2 + R y_o + S = 0 \qquad \text{obtain}$$

$$y_o = \frac{-R \pm \sqrt{R^2 - 4S}}{2} .$$

From solution sets (x_o, y_{o1}) and (x_o, y_{o2}) choose the one that best satisfies equation (II-4).

c) If $E \neq 0$ and $F \neq 0$, then

$$y_o = (F/E)x_o + (G/E) = C x_o + D.$$

Let

$$U = C^2 + 1$$
$$R = 2CD - C(y_1 + y_2) - (x_1 + x_2)$$
$$S = D^2 - D(y_1 + y_2) + y_1 y_2 + x_1 x_2 .$$

Then, from

$$U x_o^2 + R x_o + S = 0 \qquad \text{obtain}$$

$$x_o = \frac{-R \pm \sqrt{R^2 - 4US}}{2U} .$$

From solution sets (x_{o1}, y_{o1}) and (x_{o2}, y_{o2}) choose the one that best satisfies equation (II-4).

2.1.3.3) 3rd case: $\alpha_{2\ell 3} = 90^\circ$ and $\alpha_{1\ell 2} \neq 90^\circ$

Let

$$A = \tan \alpha_{1\ell 2}$$
$$E = A(y_1 - y_3) + x_1 - x_3$$
$$F = A(x_3 - x_1) + y_1 - y_2$$
$$G = A(x_1 x_2 + y_1 y_2 - x_2 x_3 - y_2 y_3) + x_1 y_2 - x_2 y_1 .$$

a) If $F = 0$, then

$$y_0 = G/E = D .$$

Let

$$R = -(x_2 + x_3)$$
$$S = D^2 - D(y_2 + y_3) + y_2 y_3 + x_2 x_3 .$$

Then, from

$$x_0^2 + R x_0 + S = 0 \qquad \text{obtain}$$

$$x_0 = \frac{-R \pm \sqrt{R^2 - 4S}}{2} .$$

From solution sets (x_{01}, y_0) and (x_{02}, y_0) choose the one that best satisfies equation (II-3).

b) If $E = 0$, then

$$x_0 = -G/F = H .$$

Let

$$R = -(y_2 + y_3)$$
$$S = H^2 - H(x_2 + x_3) + x_2 x_3 + y_2 y_3 .$$

Then, from

$$y_0^2 + R y_0 + S = 0 \qquad \text{obtain}$$

$$y_0 = \frac{-R \pm \sqrt{R^2 - 4S}}{2}.$$

From solution sets (x_0, y_{01}) and (x_0, y_{02}) choose the one that best satisfies equation (II-3).

c) If $E \neq 0$ and $F \neq 0$, then

$$y_0 = (F/E)x_0 + (G/E) = Cx_0 + D.$$

Let

$$U = C^2 + 1$$

$$R = 2CD - C(y_2 + y_3) - (x_2 + x_3)$$

$$S = D^2 - D(y_2 + y_3) + x_2 x_3 + y_2 y_3 .$$

Then, from

$$U x_0^2 + R x_0 + S = 0 \qquad \text{obtain}$$

$$x_0 = \frac{-R \pm \sqrt{R^2 - 4US}}{2U}.$$

From solution sets (x_{01}, y_{01}) and (x_{02}, y_{02}) choose the one that best satisfies equation (II-3).

2.1.3.4) 4th case: $\alpha_{1P2} = 90^\circ$ and $\alpha_{2P3} = 90^\circ$

Let

$$E = y_1 - y_3$$

$$F = x_3 - x_1$$

$$G = x_1 x_2 + y_1 y_2 - y_2 y_3 - x_2 x_3 .$$

a) If $F = 0$, then

$$y_0 = G/E = D \quad \text{and} \quad x_0 = x_1 .$$

b) If $E = 0$, then

$$x_0 = -G/F = H \quad \text{and} \quad y_0 = y_1 .$$

c) If $E \neq 0$ and $F \neq 0$, then

$$y_0 = (F/E)x_0 + (G/E) = C x_0 + D .$$

Let

$$U = C^2 + 1$$

$$R = 2CD - C(y_2 + y_3) - (x_2 + x_3)$$

$$S = D^2 - D(y_2 + y_3) + y_2 y_3 + x_2 x_3 .$$

Then, from

$$U x_0^2 + R x_0 + S = 0 \qquad \text{obtain}$$

$$x_0 = \frac{-R \pm \sqrt{R^2 - 4US}}{2U} .$$

From solution sets (x_{01}, y_{01}) and (x_{02}, y_{02}) choose the one that best satisfies

$$(y_2 - y_0)(y_1 - y_0) + (x_2 - x_0)(x_1 - x_0) = 0 . \qquad (\text{II}-5)$$

2.2) Determination of azimuths Az_{oi} between "initial point" $P_0\ (x_0\, y_0)$ and stations $S_i\ (x_i\, y_i)$

2.2.1) Two angles, Az_{oi} and $Az_{oi} + 180°$, satisfy

the equation

$$Az_{oi} = \tan^{-1} \frac{x_i - x_o}{y_i - y_o} \qquad (i = 1, 2, \ldots, N+1).$$

Also, Az_{oi} must be a positive angle between 0 and 2π. Since, in general, computers give a solution for the above equation between $(-\pi/2)$ and $(+\pi/2)$, then a criterion will be established for selecting the valid solution.

2.2.2) Criterion:

a) If $y_o = y_i$ and $x_o > x_i$, then $Az_{oi} = 3\pi/2$.

b) If $y_o = y_i$ and $x_o < x_i$, then $Az_{oi} = \pi/2$.

For $y_o \neq y_i$ designate by α_{oi} the solution given by a computer of

$$\alpha_{oi} = \tan^{-1} \frac{x_i - x_o}{y_i - y_o} \qquad (i = 1, 2, \ldots, N+1).$$

Then:

c) If $\alpha_{oi} \geqslant 0$ and $x_o > x_i$, then

$$Az_{oi} = \alpha_{oi} + \pi.$$

d) If $\alpha_{oi} \geqslant 0$ and $x_o < x_i$, then

$$Az_{oi} = \alpha_{oi}.$$

e) If $\alpha_{oi} < 0$ and $x_o > x_i$, then

$$Az_{oi} = \alpha_{oi} + 2\pi.$$

f) If $\alpha_{oi} < 0$ and $x_o < x_i$, then

$$Az_{oi} = \alpha_{oi} + \pi.$$

2.3) Determination of elements L_i of matrix L:

$$L_i = \alpha_{i\ell(i+1)} + Az_{0i} - Az_{0(i+1)} \qquad (i = 1, \ldots, N).$$

Note, L_i must be expressed in radians.

2.4) Determination of squared distances between $P_0(x_0, y_0)$ and $S_i(x_i, y_i)$:

$$(S_{0i})^2 = (x_0 - x_i)^2 + (y_0 - y_i)^2 \qquad (i = 1, \ldots, N).$$

2.5) Determination of elements a_{ij} (i=1,2,...,N; j=1,2) of matrix A:

$$a_{i1} = \frac{y_0 - y_{i+1}}{(S_{0(i+1)})^2} - \frac{y_0 - y_i}{(S_{0i})^2} \qquad (i = 1, \ldots, N)$$

$$a_{i2} = \frac{x_0 - x_i}{(S_{0i})^2} - \frac{x_0 - x_{i+1}}{(S_{0(i+1)})^2} \qquad (i = 1, \ldots, N).$$

Step 3) Normal equations

3.1) Determine matrix $A^T W$ (a matrix 2 x N).

3.2) Determine matrix $A^T W A$ (a matrix 2 x 2).

3.3) Determine matrix $(A^T W A)^{-1}$ (a matrix 2 x 2) as indicated in Step 3.3 of subsection II.A.3.a.

3.4) Determine matrix $A^T W L$ (a matrix 2 x 1).

3.5) Finally, determine

$$X = (A^T W A)^{-1} \, (A^T W L).$$

Step 4) First adjusted values

As indicated in Step 4 of subsection II.A.3.a.

Step 5) 2nd iteration

As indicated on Step 5 of subsection II.A.3.a.

Step 6) Next iterations

As indicated on Step 6 of subsection II.A.3.a.

C. FIX DETERMINATION BY TWO RANGE DISTANCES AND ONE AZIMUTH

This problem illustrates how to deal with observations of different kinds (distances and angles). The procedures for obtaining the residuals and the weight matrix are more complex.

1. Solution for Two Range Distances and One Azimuth from 3 Different Stations.

Given a positioning problem as diagramed in FIG II-7, in which:

R_1 - is the observed range distance from station #1

R_2 - is the observed range distance from station #2

A - is the observed azimuth from station #3

(x_1, y_1) - are the grid coordinates of station #1

(x_2, y_2) - are the grid coordinates of station #2

(x_3, y_3) - are the grid coordinates of station #3

σ_1 - is the standard error of R_1 (in meters)

σ_2 - is the standard error of R_2 (in meters)

σ_3 - is the standard error of A (in degrees)

the grid coordinates of vessel's position $P(x,y)$ will be determined.

Step 1) Formulation of observation equations

1.1) The analytical expression for the range distance between station i (i=1,2) and vessel's position P(xy) is given by

$$r_i \text{ (meters)} = \left[(x - x_i)^2 + (y - y_i)^2 \right]^{1/2} = F(x,y) \quad (i = 1,2).$$

The function $F(xy)$ must be expressed in a Taylor's series around an "initial position", P_o, whose coordinates are defined as x_o and y_o. Evaluating the zero and first order terms of the series, the following expression is obtained:

$$r_i = \left[(x_o - x_i)^2 + (y_o - y_i)^2\right]^{1/2} +$$

$$\frac{x_o - x_i}{\left[(x_o-x_i)^2 + (y_o-y_i)^2\right]^{1/2}} \cdot \Delta x + \frac{y_o - y_i}{\left[(x_o-x_i)^2 + (y_o-y_i)^2\right]^{1/2}} \cdot \Delta y .$$

Then, designating by s_{io} the distance from station i (i=1,2) to "initial point" P_o (x_o, y_o), the following expression is obtained:

$$S_{io} = \left[(x_o-x_i)^2 + (y_o-y_i)^2\right]^{1/2} .$$

The observation equations are given by

$$V_i = \frac{x_o - x_i}{S_{io}} \cdot \Delta x + \frac{y_o - y_i}{S_{io}} \cdot \Delta y - (R_i - S_{io}) \quad (i = 1,2)$$

where R_i is the observed range distance.

In this result it should be noted that the residuals, v_i (i=1,2), are expressed in meters.

1.2) The analytical expression for the azimuth between station 3 and $P(x,y)$ is given by

$$\mathbf{AB}\ (\text{radians}) = \tan^{-1} \frac{x - x_3}{y - y_3} .$$

Therefore, the observation equation is expressed as

$$\tan^{-1} \frac{x - x_3}{y - y_3} - A = V_3$$

where A is the observed azimuth angle. In that result, it should be noted that V_3 is expressed in radians. Therefore, it will be necessary to obtain V_3 expressed in meters. (See FIG II-5).

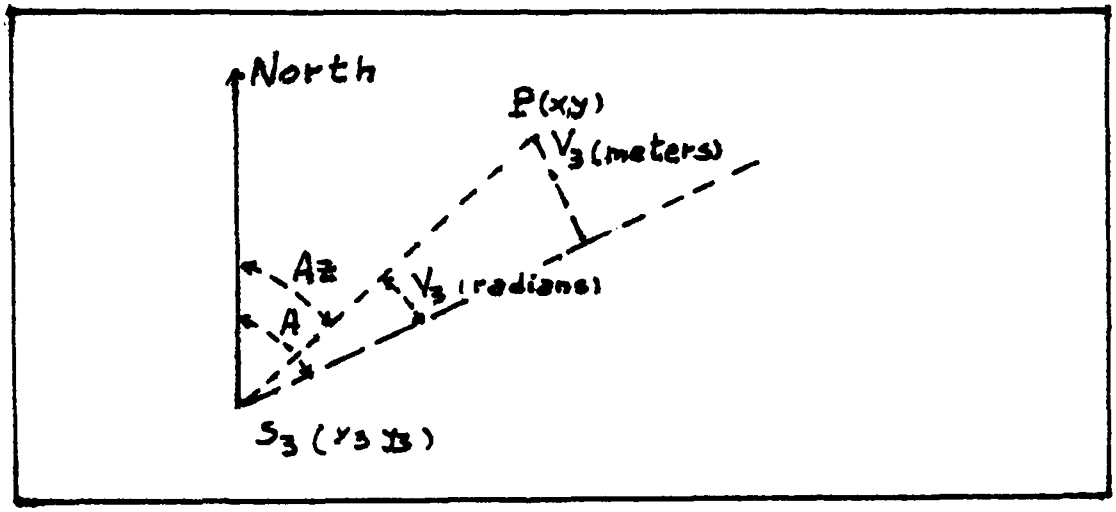


FIG II-5: CONVERTING ANGULAR RESIDUAL INTO METRICAL RESIDUAL

From the FIG II-5 is concluded that

V_3(meters) = sin V_3 (radians) x distance between P and S_3

or

$$V_3 \text{ (meters)} = \sin\left[\tan^{-1} \frac{x - x_3}{y - y_3} - A\right] \cdot \left[(x - x_3)^2 + (y - y_3)^2\right]^{1/2}$$

Expressing the function $V_3\,(xy)$ as a Taylor's series around the "initial point" $P_0\,(x_0, y_0)$, and taking only the zero and first order terms, the following is obtained:

$$V_3 = \sin\left[\tan^{-1}\frac{x_0 - x_3}{y_0 - y_3} - A\right]\cdot\left[(x_0 - x_3)^2 + (y_0 - y_3)^2\right]^{1/2} +$$

$$\left\{\cos\left[\tan^{-1}\frac{x_0 - x_3}{y_0 - y_3} - A\right]\cdot\frac{y_0 - y_3}{\left[(x_0 - x_3)^2 + (y_0 - y_3)^2\right]^{1/2}} +\right.$$

$$\left.\sin\left[\tan^{-1}\frac{x_0 - x_3}{y_0 - y_3} - A\right]\cdot\frac{x_0 - x_3}{\left[(x_0 - x_3)^2 + (y_0 - y_3)^2\right]^{1/2}}\right\}\Delta X +$$

$$\left\{\cos\left[\tan^{-1}\frac{x_0 - x_3}{y_0 - y_3} - A\right]\cdot\frac{x_3 - x_0}{\left[(x_0 - x_3)^2 + (y_0 - y_3)^2\right]^{1/2}} +\right.$$

$$\left.\sin\left[\tan^{-1}\frac{x_0 - x_3}{y_0 - y_3} - A\right]\cdot\frac{y_0 - y_3}{\left[(x_0 - x_3)^2 + (y_0 - y_3)^2\right]^{1/2}}\right\}\Delta y\,.$$

Designating by S_{30} and Az_{30} the distance and azimuth between station $S_3\,(x_3, y_3)$ and "initial point" $P_0\,(x_0, y_0)$, then

$$S_{30} = \left[(x_0 - x_3)^2 + (y_0 - y_3)^2\right]^{1/2}$$

and

$$Az_{30} = \tan^{-1}\left[\frac{x_0 - x_3}{y_0 - y_3}\right].$$

Therefore, the observation equation may be written as

$$V_3 = \sin(Az_{30} - A) \cdot S_{30} +$$

$$\left\{ \cos(Az_{30} - A) \cdot \frac{y_0 - y_3}{S_{30}} + \sin(Az_{30} - A) \cdot \frac{x_0 - x_3}{S_{30}} \right\} \Delta X +$$

$$\left\{ \cos(Az_{30} - A) \cdot \frac{x_3 - x_0}{S_{30}} + \sin(Az_{30} - A) \cdot \frac{y_0 - y_3}{S_{30}} \right\} \Delta y \; .$$

1.3) Finally, the observation equations in matrix notation are expressed as

$$AX - L = V$$

where the elements a_{ij} and l_i of matrices A and L are given by

$$a_{11} = (x_0 - x_1) / S_{10} \qquad a_{12} = (y_0 - y_1) / S_{10}$$

$$a_{21} = (x_0 - x_2) / S_{20} \qquad a_{22} = (y_0 - y_2) / S_{20}$$

$$a_{31} = \cos(Az_{30} - A) \cdot \frac{y_0 - y_3}{S_{30}} + \sin(Az_{30} - A) \cdot \frac{x_0 - x_3}{S_{30}}$$

$$a_{32} = \cos(Az_{30} - A) \cdot \frac{x_3 - x_0}{S_{30}} + \sin(Az_{30} - A) \cdot \frac{y_0 - y_3}{S_{30}}$$

$$l_1 = R_1 - S_{10} \qquad l_2 = R_2 - S_{20}$$

$$l_3 = -\sin(Az_{30} - A) \cdot S_{30}$$

Step 2) Determination of weight matrix W

The standard errors σ_1 and σ_2 of range observations are expressed in meters; the standard error σ_3 of the observed azimuth angle is expressed in degrees. Therefore, it will be necessary to obtain σ_3 expressed in meters. (See FIG II-6).

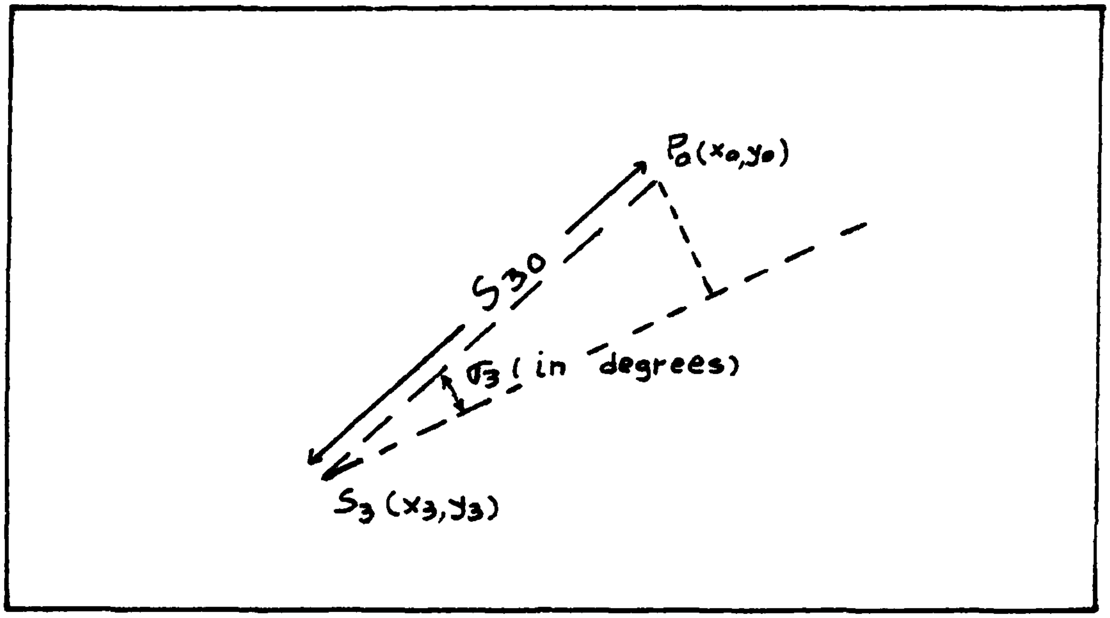


FIG II-6: CONVERTING ANGULAR STANDARD DEVIATION INTO METRICAL STANDARD DEVIATION

From FIG II-6 is concluded that

$$\sigma_3 \text{ (meters)} = \sin \sigma_3 \text{ (in degrees)} \times S_{30} .$$

Having obtained σ_3 expressed in meters, the procedure for obtaining the weight matrix W is as indicated in Step 1 of subsection II.A.3.a.

Step 3) Normal equations

Forming the normal equations, the adjusted values for Δx and Δy are given by

$$X = (A^T W A)^{-1} (A^T W L).$$

Step 4) With the values Δx and Δy a new "initial point" $P_0'(x_0', y_0')$ is obtained;

$$\begin{cases} x_0' = x_0 + \Delta x \\ y_0' = y_0 + \Delta y . \end{cases}$$

Step 5) For the new coordinates (x_0', y_0') of "initial point", the value of σ_3 (in meters) is recomputed and the weight matrix W readjusted.

Step 6) The procedure will be repeated, in an iterative way, until the increments Δx and Δy become vanishingly small, or, in practical terms, converging to within a specified tolerance.

Then, the most probable values for the coordinates (xy) will coincide with those obtained for the last "initial point".

2. Numerical Example

Referring to FIG II-7, the grid coordinates (U.T.M.) of shore stations are

COORDINATES	LUCES (#1)	MB4 (#2)	MUSSEL (#3)
x	595,794.5	603,425.2	597,967.8
y	4,055,042.7	4,053,917.2	4,053,453.2

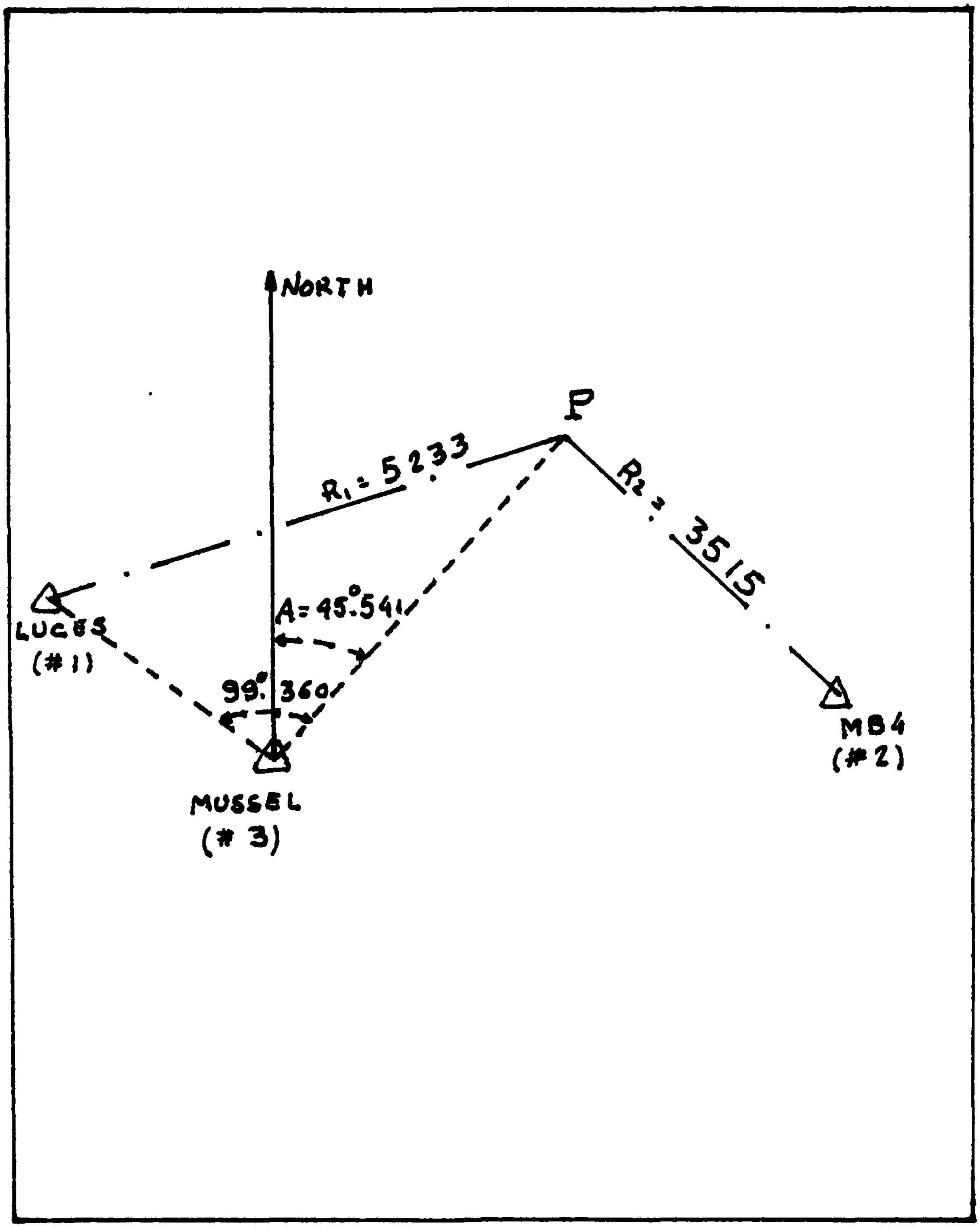


FIG II-7: FIX BY TWO RANGE DISTANCES AND ONE AZIMUTH

The following angle has been measured by theodolite:

P - MUSSEL - LUCES = $99^\circ.360$

For illustrative purposes, the instrument is assigned a standard error $\sigma_3 = 0^\circ.024$.

The following distances were measured:

P - LUCES = 5233 m

P - MB4 = 3515 m

Their standard errors are assumed to be $\sigma_1 = \sigma_2 = 10$ m (fictitious values)

Step 1) The azimuth between MUSSEL and LUCES is given by

$$Az_{31} = \tan^{-1} \frac{x_1 - x_3}{y_1 - y_3} = 306^\circ.181 .$$

Therefore, the following data are available:

$$\begin{array}{ll} R_1 = 5233 \text{ m} & \sigma_1 = 10 \text{ m} \\ R_2 = 3515 \text{ m} & \sigma_2 = 10 \text{ m} \\ A = 45^\circ.541 & \sigma_3 = 0^\circ.024 \end{array}$$

Step 2) Formulation of observation equations

2.1) Determination of first "initial point"

The first "initial point" will be the point determined by range distances R_1 and R_2 (for which the azimuth from station 3 is closer to A). Therefore, the point $P_0(x_0, y_0)$ will satisfy the following system of equations:

$$\begin{cases} (x_0 - x_1)^2 + (y_0 - y_1)^2 = R_1^2 \\ (x_0 - x_2)^2 + (y_0 - y_2)^2 = R_2^2 . \end{cases}$$

Introducing numerical values, the following solution sets for the above equations are obtained :

$$\begin{cases} x_{01} = 600,867.2 \\ y_{01} = 4,056,328.0 \end{cases} \quad \text{and} \quad \begin{cases} x_{02} = 600,280.1 \\ y_{02} = 4,052,347.6 \end{cases}.$$

The azimuth from station #3 to (x_{01}, y_{01}) and (x_{02}, y_{02}), respectively, are obtained: $Az_{301} = 45.^\circ 24$ and $Az_{302} = 115^\circ.5$. Therefore, the valid solution is the one corresponding to Az_{301} , i.e.,

$$\begin{cases} x_0 = x_{01} = 600,867.2 \\ y_0 = y_{01} = 4,056,328.0 \end{cases}$$

2.2) Determination of azimuth between station 3 and $P_0\,(x_0, y_0)$:

$$Az_{30} = \tan^{-1} \frac{x_0 - x_3}{y_0 - y_3} = 45.^\circ 244$$

Then, $Az_{30} - A = -0.^\circ 297$.

2.3) Determining distances between stations and P_0,

$$S_{10} = \left[(x_1 - x_0)^2 + (y_1 - y_0)^2 \right]^{1/2} = 5233.0$$

$$S_{20} = \left[(x_2 - x_0)^2 + (y_2 - y_0)^2 \right]^{1/2} = 3515.0$$

$$S_{30} = \left[(x_3 - x_0)^2 + (y_3 - y_0)^2 \right]^{1/2} = 4083.0 .$$

2.4) Therefore, the elements a_{ij} and L_i of matrices A and L will be

$$a_{11} = (x_0 - x_1)/S_{10} = 0.969\,368 \qquad a_{21} = (y_0 - y_1)/S_{10} = 0.245\,614$$

$$a_{21} = (x_0 - x_2)/S_{20} = -0.727\,738 \qquad a_{22} = (y_0 - y_2)/S_{20} = 0.685861$$

$$a_{31} = \cos(Az_{30} - A) \cdot \frac{y_0 - y_3}{S_{30}} + \sin(Az_{30} - A) \cdot \frac{x_0 - x_3}{S_{30}} = 0.700\,400$$

$$a_{32} = \cos(Az_{30} - A) \cdot \frac{x_3 - x_0}{S_{30}} + \sin(Az_{30} - A) \cdot \frac{y_0 - y_3}{S_{30}} = -0.713\,755$$

$$L_1 = R_1 - S_{10} = 0$$

$$L_2 = R_2 - S_{20} = 0$$

$$L_3 = -\sin(Az_{30} - A) \cdot S_{30} = 21.165$$

2.5) The observation equations in matrix notation may be written as

$$\begin{bmatrix} 0.969\,368 & 0.245\,614 \\ -0.727\,738 & 0.685\,861 \\ 0.700\,400 & -0.713\,755 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 21.165 \end{bmatrix} = \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix}$$

Step 3) Normal equations

3.1) Determination of weight matrix W:

$$\sigma_1 = 10 \text{ m}$$
$$\sigma_2 = 10 \text{ m}$$
$$\sigma_3 = 0^\circ.024 \quad \Rightarrow \quad \sigma_3 \text{ (meters)} = S_{30} \cdot \sin(0^\circ.024)$$
$$= 1.710 \text{ m}$$

Then

$$1/\sigma_1^2 = 0.01$$
$$1/\sigma_2^2 = 0.01$$
$$1/\sigma_3^2 = 0.34 .$$

Setting the least weight equal to one, it will be obtained that

$$\omega_1 = 1 \qquad\qquad W = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 34 \end{bmatrix} .$$
$$\omega_2 = 1 \qquad \text{or}$$
$$\omega_3 = 34$$

3.2) The solution of normal equations is given by

$$X = (A^T W A)^{-1} (A^T W L).$$

3.2.1) Determination of $A^T W$:

$$A^T W = \begin{bmatrix} 0.969\,368 & -0.727738 & 23.813\,600 \\ 0.245\,614 & 0.685861 & -24.267670 \end{bmatrix} .$$

3.2.2) Determination of $A^T W A$:

$$A^T W A = \begin{bmatrix} 18.148 & -17.258 \\ -17.258 & 17.852 \end{bmatrix}.$$

3.2.3) Determination of $(A^T W A)^{-1}$:

$$(A^T W A)^{-1} = \begin{bmatrix} 0.68295 & 0.66023 \\ 0.66023 & 0.69427 \end{bmatrix}.$$

3.2.4) Determination of $(A^T W L)$:

$$(A^T W L) = \begin{bmatrix} 504.015 \\ -513.625 \end{bmatrix}.$$

3.2.5) Finally,

$$X = \begin{bmatrix} 5.1 \\ -23.8 \end{bmatrix}.$$

Step 4) First adjusted values

With the values ΔX =5.1 and Δy = - 23.8 a new "initial point" is obtained:

X_o = 600,867.2 + 5.1 = 600,872.3

y_o = 4,056,328.0 - 23.8 = 4,056,304.2

Step 5) With the new values for the "initial point" the procedure indicated in steps 2.2, 2.3, 2.4, 2.5, 3 and 4 is repeated and a "closer" initial point is obtained.

Step 6) That procedure must be repeated, in an iterative way, until the increments Δx and Δy become vanishingly small, or, in practical terms, converge to within a specified tolerance. Then, the last "initial point" obtained will coincide with the most probable position for P.

These computations may be compared with those shown in the computer output section on page 150. Differences in the results are due to the fact that the calculations illustrated on the preceeding pages were only carried out for one iteration.

3. <u>Solution for the General Case</u>

The solution will be presented in such a way that easily can be implemented by an algorithm satisfying a modular design.

a. Two cases will be considered:

1st case) Two range distances and one azimuth from three stations (See FIG. II-8)

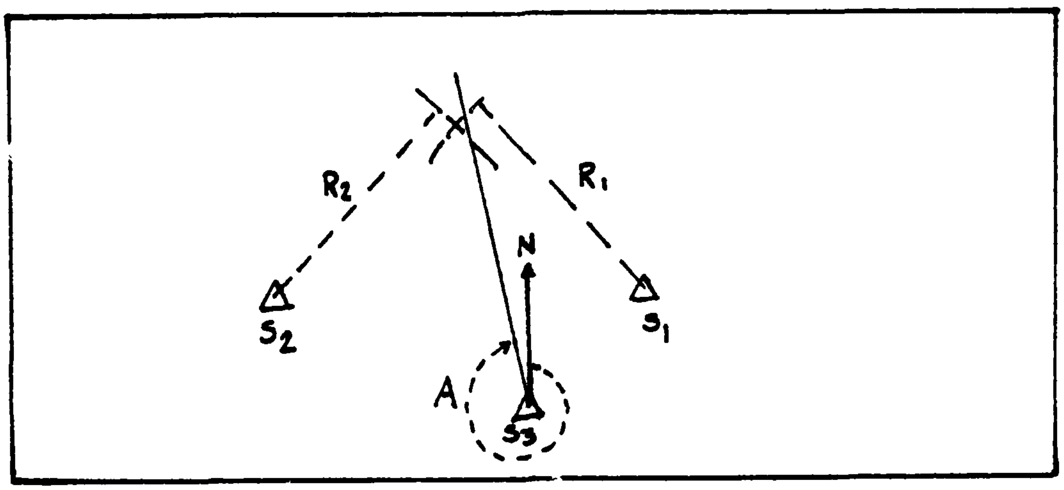


FIG. II-8: FIX FROM 3 STATIONS

Designate by S_1 and S_2 the stations from which range distances are observed; the station from which an azimuth is observed will be designated by S_3.

2nd case) Two range distances and one azimuth from just two stations. (See FIG II-9)

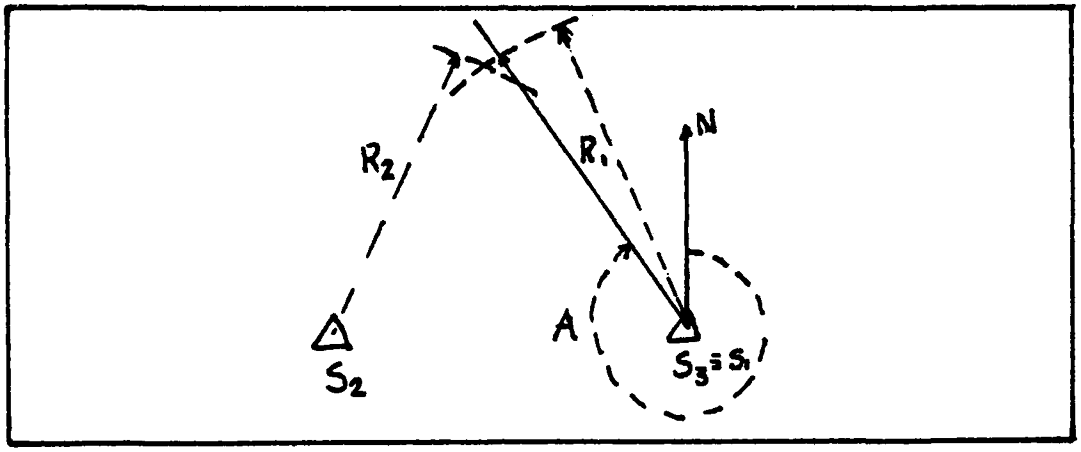


FIG II-9: FIX FROM 2 STATIONS

Designate by S_1 the station from whch an azimuth and a range distance were measured (S_1 concides with S_3); the remaining station will be designated by S_2.

b. Given

(x_1, y_1) - grid coordinates of station S_1
(x_2, y_2) - grid coordinates of station S_2
(x_3, y_3) - grid coordinates of station S_3
(in the 2nd case, they coincide with the coordinates (x_1, y_1) of S_1)
R_1 - range distance from station S_1
R_2 - range distance from station S_2
A - azimuth from station S_3

and

σ_1 - standard error of R_1 (in meters)

σ_2 - standard error of R_2 (in meters)

σ_3 - standard error of A (in degrees)

the coordinates of vessel's position P(x,y) will be determined.

Step 1) Formulation of observation equations

1.1) Determination of first "initial point" P_o (x_o, y_o)

The first "initial point" will be:

a) one of the intersection points of circumferences centered at S_1 and S_2 with radius ranges of R_1 and R_2 respectively.

b) the intersection point which lies closer to the azimuth line through S_3.

Therefore, the following equations must be satisfied:

$$\begin{cases} (x_1 - x_o)^2 + (y_1 - y_o)^2 = R^2 \\ (x_2 - x_o)^2 + (y_2 - y_o)^2 = R_2^2 . \end{cases}$$

Let

$$E = R_1^2 - R_2^2 + y_2^2 - y_1^2 + x_2^2 - x_1^2 .$$

Then,

$$x_o = \frac{E + 2\,(y_1 - y_2)\,y_o}{2\,(x_2 - x_1)} .$$

1.1.1) For $X_2 \neq X_1$ proceed as follows:

Let

$$E_1 = \left(\frac{y_1 - y_2}{x_2 - x_1} \right)^2 + 1$$

$$E_2 = \frac{E\,(y_1 - y_2)}{(x_2 - x_1)^2} - 2\,x_1 \left(\frac{y_1 - y_2}{x_2 - x_1} \right) - 2\,y_1$$

$$E_3 = \left(\frac{E}{2\,(x_2 - x_1)} \right)^2 - \frac{x_1 E}{x_2 - x_1} - R_1^2 + x_1^2 + y_1^2$$

$$E_4 = (E_2)^2 - 4\,(E_1)\,(E_3)\,.$$

Then, $$y_o = \frac{-(E_2) \pm \sqrt{E_4}}{2\,E_1}\,.$$

a) If $E_4 < o$, then the two circumferences don't intersect; therefore, choose point Q as first "Initial point" (See FIG II-10).

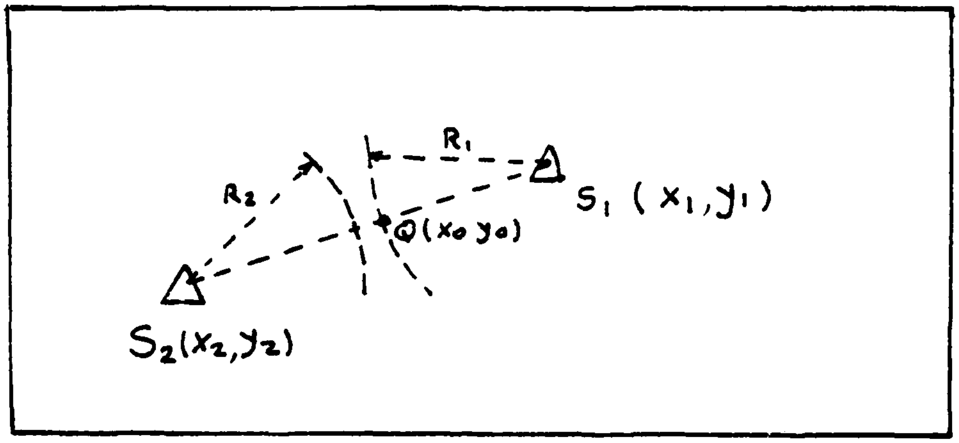


FIG II-10: RANGE DISTANCES NOT INTERSECTING

Then,

$$\begin{cases} x_0 = x_1 + \dfrac{R_1 \; (x_2 - x_1)}{\left[(x_2-x_1)^2 + (y_2-y_1)^2\right]^{1/2}} \\[2ex] y_0 = y_1 + \dfrac{R_1 \; (y_2 - y_1)}{\left[(x_2-x_1)^2 + (y_2-y_1)^2\right]^{1/2}} \end{cases} .$$

b) If $E_4 = 0$, then the two circumferences are tangent. Therefore

$$\begin{cases} y_0 = - \dfrac{E_2}{2 E_1} \\[2ex] x_0 = \dfrac{E}{2\,(x_2 - x_1)} + \dfrac{y_1 - y_2}{x_2 - x_1}\, y_0 \end{cases} .$$

c) If $E_4 > 0$, then the two circumferences intersect at 2 points; therefore, the following intersection points are obtained:

$$\begin{cases} y_{01} = \left[- E_2 + \sqrt{E_4} \right] / 2 E_1 \\[2ex] x_{01} = \dfrac{E}{2(x_2 - x_1)} + \dfrac{y_1 - y_2}{x_2 - x_1}\, y_{01} \end{cases}$$

and

$$\begin{cases} y_{02} = \left[- E_2 - \sqrt{E_4} \right] / 2 E_1 \\[2ex] x_{02} = \dfrac{E}{2(x_2 - x_1)} + \dfrac{y_1 - y_2}{x_2 - x_1}\, y_{02} \end{cases}$$

and

$$y_{02} = \left[-E_2 - \sqrt{E_4}\right]/2E_1$$

$$x_{02} = E/\left[2(x_2-x_1)\right] + \left[(y_1-y_2)/(x_2-x_1)\right] y_{02}.$$

c.1) If $x_3 = x_{01}$ and $y_3 = y_{01}$, then

$$\begin{cases} x_0 = x_{02} \\ y_0 = y_{02}. \end{cases}$$

c.2) If $x_3 = x_{02}$ and $y_3 = y_{02}$, then

$$\begin{cases} x_0 = x_{01} \\ y_0 = y_{01}. \end{cases}$$

c.3) Ôtherwise, determine azimuths between station $S_3(x_3, y_3)$ and (x_{0i}, y_{0i}) (for i=1,2).

Criterion:

I) If $y_{0i} = y_3$ and $x_{0i} > x_3$, then

$$Az_{30i} = \pi/2.$$

II) If $y_{0i} = y_3$ and $x_{0i} < x_3$, then

$$Az_{30i} = 3\pi/2.$$

For $y_{0i} \neq y_3$ designate by α_{3i} the solution given by a computer of

$$\alpha_{3i} = \tan^{-1} \frac{x_{0i} - x_3}{y_{0i} - y_3} \quad (i = 1,2).$$

Then:

III) If $\alpha_{3i} \geq 0$ and $x_{oi} \geq x_3$, let

$$Az_{30i} = \alpha_{3i}.$$

IV) If $\alpha_{3i} < 0$ and $x_{oi} < x_3$, let

$$Az_{30i} = \alpha_{3i} + 2\pi.$$

V) Otherwise, $Az_{30i} = \alpha_{3i} + \pi$.

Having determined azimuths Az_{301} and Az_{302}, check which one is closer to the observed azimuth A.

I) If $Az_{301} = Az_{302} = A$, then the solution will be undetermined (See FIG II-11).

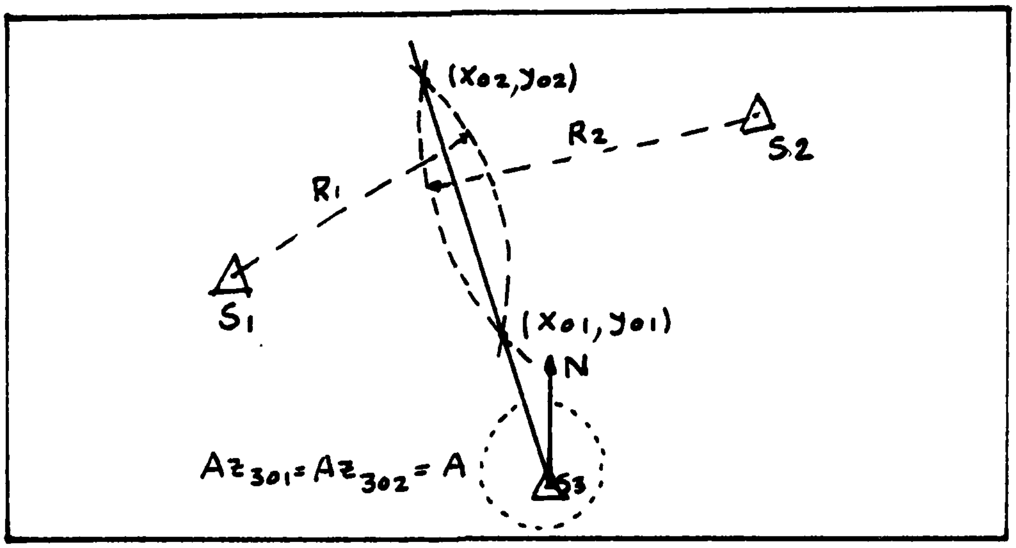


FIG II-11: UNDETERMINED FIX BY 2 RANGE DISTANCES AND 1 AZIMUTH

II) If $(Az_{301}-A)^2 = (Az_{302}-A)^2$, then choose

$$\begin{cases} x_0 = (x_{01} + x_{02}) / 2 \\ y_0 = (y_{01} + y_{02}) / 2 . \end{cases}$$

III) If $(Az_{301}-A)^2 > (Az_{302}-A)^2$, then choose

$$\begin{cases} x_0 = x_{02} \\ y_0 = y_{02} . \end{cases}$$

IV) If $(Az_{301}-A)^2 < (Az_{302}-A)^2$, then choose

$$\begin{cases} x_0 = x_{01} \\ y_0 = y_{01} . \end{cases}$$

1.1.2) For $x_2 = x_1$,

the following equation is obtained:

$$y_0 = E / [2 (y_2 - y_1)] .$$

Let

$$F = R_1^2 - (y_1 - y_0)^2 .$$

Then,

$$x_0 = x_1 \pm \sqrt{F} .$$

a) If $F \leq 0$, the two circumferences do not intersect or are tangent; then

$$x_0 = x_1 .$$

b) If $F > 0$, the two circumferences intersect at points (x_{01}, y_{01}) and (x_{02}, y_{02}) ; then

$$\begin{cases} x_{01} = x_1 + \sqrt{F} \\ y_{01} = y_0 \end{cases}$$

and

$$\begin{cases} x_{02} = x_1 - \sqrt{F} \\ y_{02} = y_0 \end{cases} .$$

b.1) If $x_3 = x_{01}$ and $y_3 = y_{01}$, then

$$x_0 = x_{02}$$

b.2) If $x_3 = x_{02}$ and $y_3 = y_{02}$, then

$$x_0 = x_{01}$$

b.3) Otherwise, determine azimuths Az_{301} and Az_{302} between station $S_3\ (x_3, y_3)$ and (x_{0i}, y_{0i}) (i=1,2) using criterion presented in step 1.1.1 . Having determined Az_{301} and Az_{302}, determine which one is closer to A .

I) If $Az_{301} = Az_{302} = A$, then the solution is undetermined.

II) If $(Az_{301}-A)^2 = (Az_{302}-A)^2$,

then choose

$$x_0 = x_1 .$$

III) If $(Az_{301}-A)^2 > (Az_{302}-A)^2$,

then choose

$$x_0 = x_{02} .$$

IV) If $(Az_{301}-A)^2 < (Az_{302}-A)^2$,

then choose

$$x_0 = x_{01} .$$

1.2) Determining the azimuth Az_{30} between station $S_3(x_3, y_3)$ and (x_0, y_0)

Criterion:

I) If $y_0 = y_3$ and $x_0 > x_3$, then

$$Az_{30} = \pi / 2 .$$

II) If $y_0 = y_3$ and $x_0 < x_3$, then

$$Az_{30} = 3\pi / 2 .$$

For $y_0 \neq y_3$ designate by α_3 the solution given by a computer of

$$\alpha_3 = \tan^{-1} \frac{x_0 - x_3}{y_0 - y_3} .$$

Then:

III) If $\alpha_3 \geq 0$ and $x_0 \geq x_3$, then $Az_{30} = \alpha_3$.

IV) If $\alpha_3 < 0$ and $x_0 < x_3$, then $Az_{30} = \alpha_3 + 2\pi$.

V) Otherwise, $Az_{30} = \alpha_3 + \pi$.

1.3) Determining distances between stations and P_0,

$$S_{10} = [(x_1 - x_0)^2 + (y_1 - y_0)^2]^{1/2}$$
$$S_{20} = [(x_2 - x_0)^2 + (y_2 - y_0)^2]^{1/2}$$
$$S_{30} = [(x_3 - x_0)^2 + (y_3 - y_0)^2]^{1/2}.$$

1.4) Determining elements of matrix A,

$$a_{11} = \frac{x_0 - x_1}{S_{10}} \qquad a_{12} = \frac{y_0 - y_1}{S_{10}}$$

$$a_{21} = \frac{x_0 - x_2}{S_{20}} \qquad a_{22} = \frac{y_0 - y_2}{S_{20}}$$

$$a_{31} = \cos(Az_{30} - A) \cdot \frac{y_0 - y_3}{S_{30}} + \sin(Az_{30} - A) \cdot \frac{x_0 - x_3}{S_{30}}$$

$$a_{32} = \cos(Az_{30} - A) \cdot \frac{x_3 - x_0}{S_{30}} + \sin(Az_{30} - A) \cdot \frac{y_0 - y_3}{S_{30}}$$

1.5) Determining elements of matrix L,

$$l_1 = R_1 - S_{10} \qquad l_2 = R_2 - S_{20}$$
$$l_3 = \sin(A - Az_{30}) \cdot S_{30}$$

Step 2) Solution of normal equations

2.1) Obtain weight matrix W.

2.1.1) Determine standard error of observed azimuth angle expressed in meters,

$$\sigma_3 \text{ (meters)} = \sin \sigma_3 \text{ (radians)} \times S_{30}.$$

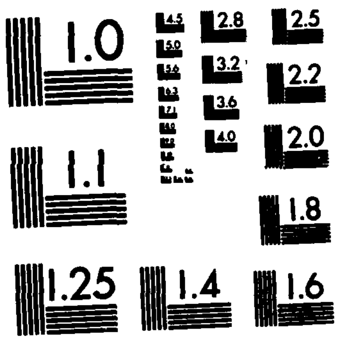
AD-A123 917 CALCULATION OF HYDROGRAPHIC POSITION DATA BY LEAST SQUARES ADJUSTMENT(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA F CASTRO E SILVA JUN 82 2/2

UNCLASSIFIED F/G 12/1 NL

END

FILMED

DTIC



2.1.2) With σ_3 expressed in meters, the procedure for obtaining the weight matrix W is as indicated in Step 1 of subsection II.A.3.a

2.1.3) Finally, σ_3 is again expressed in radians;

$$\sigma_3(\text{radians}) = \sin^{-1}\left[\sigma_3(\text{meters}) / S_{30}\right].$$

2.2) Determine matrix $A^T W$.

2.3) Determine matrix $A^T W A$.

2.4) Determine matrix $(A^T W A)^{-1}$.

2.5) Determine matrix $A^T W L$.

2.6) Finally, determine $X = (A^T W A)^{-1} (A^T W L)$.

Step 3) First adjusted values

With the values Δx and Δy obtain new "initial point" $P'_o\ (x'_o\ y'_o)$;

$$\begin{cases} x'_o = x_o + \Delta x \\ y'_o = y_o + \Delta y . \end{cases}$$

Step 4) 2nd iteration

For obtaining a "closer" initial point,repeat steps 1.2, 1.3, 1.4, 1.5, 2 and 3.

Step 5) Next iterations

Repeat step 4 until Δx and Δy become vanishingly small, or, in practical terms, converge to within a specified tolerance. Then, the adjusted values for x and y will coincide with the coordinates of the last "initial point" obtained.

III. RESULTS AND CONCLUSIONS

A. RESULTS

From the general case solutions developed for the selected positioning methods, algorithms were written in a structured programming format. All algorithms are presented in appendix I.

These modular algorithms were translated into Fortran language for implementation on the NPS computer, an IBM 3033. Program listings are provided in the Computer Programs Section beginning on page 151.

Data sets given in each Numerical Example section were input into the corresponding computer program, and the output of each run is given in the Computer Output Section starting on page 148.

Additionally, the programs were tested using several fictitious data sets to insure their performance in handling the various initial conditions which were modeled for each fixing method.

In applying these programs to real positioning data the following points should be considered:

1. The presence of blunders and systematic errors in the observations will be reflected in the dimensions of the error ellipse. If all blunders are removed by careful editing and all systematic errors are eliminated by modeling or calibration, then the size of the error ellipse will

represent the positioning error due to net geometry and random errors.

2. When the information about the standard errors of the observations is reliable (for example, determined by field calibration procedures), then the estimates obtained for the standard deviations of the observed values will be close to the a priori values (see example of fix by 3 azimuth angles in Computer Output section on page 148).

3. When no a priori values are given for the standard deviations of the observations (it is assumed that the observations are equally weighted), then the application of the least squares method will provide estimates of instrument (or observation) accuracy (see example in computer output section, on page 149).

4. When the correlation coefficient is close to one, the error ellipse becomes flatter approaching a straight line (see example of fix by two range distances and one azimuth angle in the Computer Output section on page 150).

5. When the correlation coefficient is negative, the major axis of the error ellipse runs through the 2nd and 4th quadrants. Thus, the angle from the x-axis to the major axis measured counterclockwise lies between 90° and 180°. If the correlation coefficient is positive, the major axis runs through the 1st and 3rd quadrants, and the angle from x-axis to the major axis measured counterclockwise is between 0° and 90°.

B. CONCLUSIONS

The most significant result of this thesis is that well documented programs are now available which can be used for the analysis of hydrographic positioning data. These programs may be employed to process and analyze hydrographic survey data that have been collected using one of the three positioning methods discussed. Ideally, such software should be adapted to run in a mini computer aboard a survey vessel or launch. This capability would allow "real time" analysis of positioning accuracy.

In addition to processing actual survey data, the programs may assist in survey planning. By scaling observations from existing charts of a survey area, sample data sets may be formed to test net geometry. This information can be used to establish the best location for shore control stations.

The programs are written in modular form so that they may be adapted for use by other types of positioning systems. The significant differences between all the programs lie in the modules dealing with the computation of the "initial point" and formulation of the observation equations.

It should be noted that the accuracy of the geodetic control stations has not been specifically considered in these formulations. However, any survey error in the station coordinates will be reflected in the dimensions of the error ellipse of the adjusted hydrographic position.

All of the programs were developed using a plane coordinate system model. Thus, they are primarily applicable to nearshore hydrographic positioning problems. Application to offshore hydrography would require a geodetic coordinate system model based on a selected spheroidal datum surface. Obviously, the use of a geodetic coordinate system would yield more complex analytical expressions relating the unknowns. But, once these were obtained and linearized, then the procedures for computing adjusted survey coordinates and the statistical values defining their precision are identical to those developed in this thesis.

Whether the existing programs are used in their current form or modified to accomodate other variables, one final point should be made. The most significant contribution of the least squares method to hydrographic position adjustment is its ability to quantify errors statistically. When programs are operated aboard the survey vessel in "real time ", relative accuracy achieved with conventional survey methods is elevated to absolute accuracy if redundant observations are made and adjusted using least squares.

Monitoring the size and orientation of the error ellipse alerts the user to the presence of gross blunders and inordinately large systematic errors. The need for electronic positioning system calibration can be realistically evaluated, and calibration may be performed on an as needed basis. With

sufficient redundant observations, electronic positioning systems can, in fact, become self calibrating.

As the trends in electronic and computer technology continue to decrease the cost of collecting and processing redundant observations, conventional two LOP's survey positioning will be relegated to the historical equivalent of lead line hydrography.

APPENDIX A. LEAST SQUARES PRINCIPLE AND NORMAL DISTRIBUTION

When measuring a parameter, the outcomes of that experiment can be considered as values assumed by a random variable following a normal distribution. For a random variable X following a normal distribution, the value most likely to occur is its <u>mean</u> μ_x. The true value, from a deterministic point of view, of an observed parameter is, in a stocastical sense the mean of the random variable associated with the experiment. Therefore, when using the least squares technique for the adjustment of a redundant number of observations, not only a set of "consistent" values are obtained but also the most probable values for the <u>means</u> of the random variables considered. Therefore, the adjusted values are also the best estimates for the "true" values of the parameters considered.

1. Normal distribution

The density function associated with a random variable X following a normal distribution is expressed by (see FIG A-1).

$$f_X(x) = \frac{1}{\sigma_x \sqrt{2\pi}} e^{-\left[(x-\mu_x)^2 / 2\sigma_x^2\right]}$$

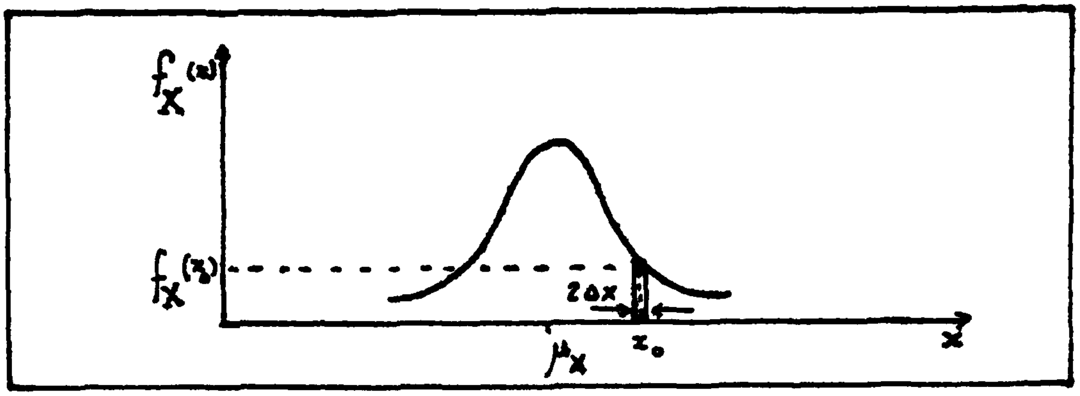


FIG A-1: NORMAL DISTRIBUTION

Then, the probability of occurrence of values between $x_0 - \Delta x$ and $x_0 + \Delta x$ will be given by

$$\int_{x_0-\Delta x}^{x_0+\Delta x} f_X(x)\, dx \, .$$

Therefore, it can be concluded that the probability of occurrence of values "around" x_0 is proportional to the density function value at that point, i.e.,

$$P\left\{ x_0 - \Delta x \leq X \leq x_0 + \Delta x \right\} = K\, f_X(x_0) \, . \quad (A-1)$$

2. Probability of occurrence of a set of values assumed by independent random variables

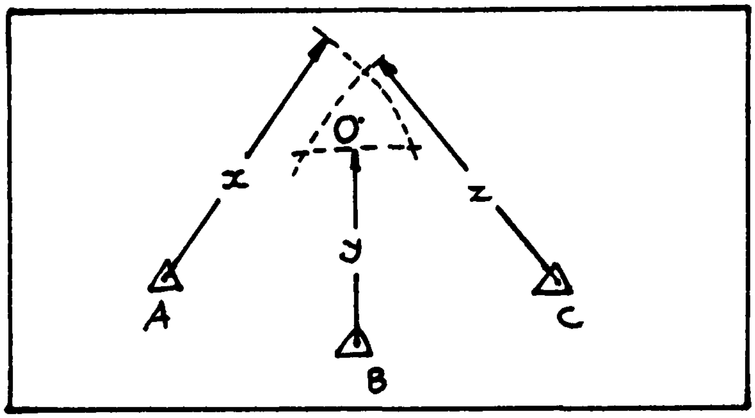


FIG A-2: FIX BY 3 RANGE DISTANCES

Suppose that the distances between a vessel and stations A, B and C are measured and the results are, respectively, x, y, and z (see FIG A-2).

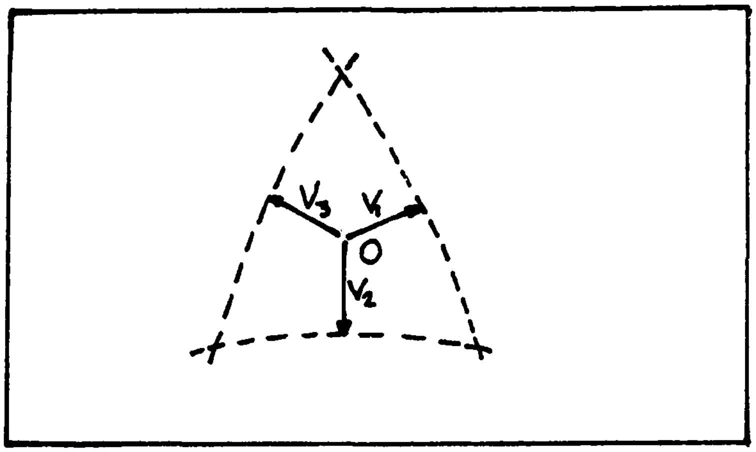


FIG A-3: RESIDUALS

If the vessel is situated at O, then the means of the random variables X, Y and Z, associated with the range distances AO, BO and CO, are at distances V_1,V_2 and V_3 from, respectively, observed values x, y and z. (See FIGS A-3 and A-4).

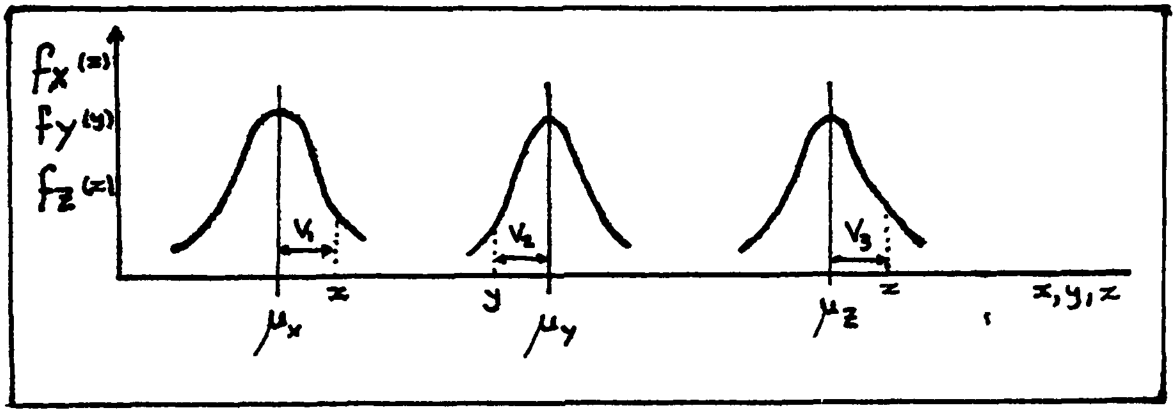


FIG A-4: RESIDUALS AND NORMAL DISTRIBUTION

The probability of occurrence of a set of observed values $\{x\ y\ z\}$ is given by

$$P(\{x\ y\ z\}) = P_1(\{x\}) \cdot P_2(\{y\}) \cdot P_3(\{z\}).$$

If the observations are equally weighted then the standard deviations σ_x, σ_y and σ_z have the same value, say, σ. Then, from (A-1), it will be obtained

$$P(\{x\ y\ z\}) = (K/\sigma\sqrt{2\pi})^3\, e^{-\left[\frac{1}{2\sigma^2}\left[(x-\mu_x)^2 + (y-\mu_y)^2 + (z-\mu_z)^2\right]\right]} \qquad (A-2)$$

Recalling that

$$V_1 = \mu_x - x$$
$$V_2 = \mu_y - y$$
$$V_3 = \mu_z - z,$$

it will be obtained from (A-2)

$$P(\{x\ y\ z\}) = \text{CONSTANT} \times e^{-\frac{1}{2\sigma^2}\left[V_1^2 + V_2^2 + V_3^2\right]}. \qquad (A-3)$$

The means μ_x, μ_y and μ_z are values such that the observed values x, y and z have a probability as high as can be

expected to occur.

Therefore, from (A-3) it will be concluded that the residual values maximizing the probability of occurrence of event $\{x\,y\,z\}$ will be the set of values minimizing the expression ($V_1^2 + V_2^2 + V_3^2$), i.e., those that minimize the sum of the squared residuals. That is the reason why the least squares technique yields the most probable values for the <u>means</u> of the random variables considered, i.e., the best estimates for the "true" values of parameters being observed.

APPENDIX B. LEAST SQUARES PRINCIPLE FOR WEIGHTED OBSERVATIONS

1. If an observed value x_i has the weight ω, then the observed value x_i is worth as much as ω observed values x_i with weight equal to unity.

A usual criterion for establish weights is to consider the weights inversely proportional to the squared standard deviations, i.e.,

$$\omega_i = \frac{1}{\sigma_i^2} .$$

Then, given a set of observed values x_i (i=0,1,2,...), unequally precise, with standard deviations σ_i, if the least precise value x_0 is considered to have a weight equal to unity (ω_0=1), the different weights ω_i will satisfy

$$\omega_i = \frac{\sigma_0^2}{\sigma_i^2} \qquad (i = 0, 1, 2, \ldots).$$

2. A set of observed values x_1, x_2, ..., x_n, with respective weights equal to ω_1, ω_2, ..., ω_n, is equal to a set of ω_1 values equal to x_1, ω_2 values equal to x_2, ..., and ω_n values equal to x_n, all with unity weight. Therefore, the sum of the squared residuals will be given by

$$\omega_1 (\mu_{x_1} - x_1)^2 + \omega_2 (\mu_{x_2} - x_2)^2 + \ldots + \omega_n (\mu_{x_n} - x_n)^2,$$

and the basic least squares principle will be expressed as

$$\sum_{i=1}^{n} \omega_i (\mu_{x_i} - x_i)^2 = \sum_{i=1}^{n} \omega_i V_i^2 = \text{minimun}$$

or, in matrix form, as

$$V^T W V = \text{minimum}$$

where

$$\begin{bmatrix} \omega_1 & 0 & 0 & \cdots & 0 \\ 0 & \omega_2 & 0 & \cdots & 0 \\ 0 & 0 & \omega_3 & \cdots & \vdots \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & \cdots & \cdots & \omega_n \end{bmatrix} .$$

APPENDIX C. NORMAL EQUATION IN ALGEBRAIC NOTATION

Below are the observation equations in algebraic notation:

$$\begin{cases} V_1 = a_{11}x_1 + a_{12}\,x_2 + \ldots + a_{1m}x_m - L_1 = 0 \\ V_2 = a_{21}x_1 + a_{22}\,x_2 + \ldots + a_{2m}x_m - L_2 = 0 \\ \text{-----------------------------} \\ V_n = a_{n1}x_1 + a_{n2}\,x_2 + \ldots + a_{nm}x_m - L_n = 0 \end{cases}$$

The unknown values $x_1, x_2, \ldots, x_m$ that satisfy the basic least squares principle

$$\sum_{i=1}^{n} \omega_i V_i^2 = \text{minimun}$$

are those that satisfy the following expression:

$$\begin{aligned} &\omega_1(a_{11}x_1 + a_{12}x_2 + \ldots + a_{1m}x_m - L_1)^2 + \\ &\omega_2(a_{21}x_1 + a_{22}x_2 + \ldots + a_{2m}x_m - L_2)^2 + \\ &\text{--------------------------} + \\ &\omega_n(a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nm}x_m - L_n)^2 = F(x_1 x_2 \ldots x_m) = \\ &= \text{minimun}. \end{aligned}$$

The values $x_1, x_2, \ldots x_m$ minimizing $F(x_1, x_2, \ldots x_m)$ are those such that

$$\frac{\partial F}{\partial x_j} = 0 \qquad (j = 1, 2, \ldots, m).$$

Considering that

$$\frac{\partial F}{\partial x_1} = 2\,[w_i a_{i1}^2]x_1 + \ldots + 2[w_i a_{i1} a_{im}]x_m - 2[w_i a_{i1} l_i] = 0$$

$$\frac{\partial F}{\partial x_2} = 2\,[w_i a_{i1} a_{i2}]x_1 + \ldots + 2\,[w_i a_{i2} a_{im}]\,x_m - 2\,[w_i a_{i2} l_i] = 0$$

$$- - - - - - - - - - - - - - - - - -$$

$$\frac{\partial F}{\partial x_m} = 2\,[w_i a_{i1} a_{im}]x_1 + \ldots + 2\,[w_i a_{im}^2]\,x_m - 2\,[w_i a_{im} l_i] = 0$$

the following normal equations are obtained :

$$\begin{cases} [w_i a_{i1}^2]\,x_1 + \ldots + [w_i a_{i1} a_{im}]\,x_m - [w_i a_{i1} l_i] = 0 \\ [w_i a_{i1} a_{i2}]\,x_1 + \ldots + [w_i a_{i2} a_{im}]\,x_m - [w_i a_{i2} l_i] = 0 \\ - - - - - - - - - - - - - - - - - \\ [w_i a_{i1} a_{im}]\,x_1 + \ldots + [w_i a_{im}^2]\,x_m - [w_i a_{im} l_i] = 0 \end{cases}$$

APPENDIX D. NORMAL EQUATIONS IN MATRIX NOTATION

Taking the observation equations in its matrix notation,

$$AX - L = V,$$

the unknown vector X satisfying the basic least squares principle,

$$V^T W V = \text{minimun},$$

is the one such that

$$(AX - L)^T W (AX - L) =$$

$$X^T A^T W A X - 2 X^T A^T W L + L^T W L = \text{minimun}.$$

The vector X satisfying the above expression will be such that

$$\frac{\partial}{\partial X}(X^T A^T W A X - 2 X^T A^T W L + L^T W L) = 0 \quad . \qquad (D-1)$$

Considering that

$$\frac{\partial}{\partial x}(X^T A^T W A X) = 2 X^T A^T W A$$

$$\frac{\partial}{\partial x}(X^T A^T W L) = L^T W^T A$$

$$\frac{\partial}{\partial x}(L^T W L) = 0,$$

it will be obtained from (D-1) that

$$2 X^T A^T W A - 2 L^T W^T A = 0.$$

Therefore, the normal equations are of the form

$$(A^T W A) X = A^T W L,$$

and the solution will be given by

$$X = (A^T W A)^{-1} (A^T W L).$$

APPENDIX E. A COMPUTATIONAL CHECK FOR THE LEAST SQUARES Adjustment Technique

By taking the normal equations

$$(A^T W A) X = A^T W L \qquad (E-1)$$

and recalling that

$$A X = V + L,$$

then the following can be obtained:

$$(A^T W A) X = A^T W V + A^T W L. \qquad (E-2)$$

Therefore, from (E-1) and (E-2) it is obtained that

$$A^T W V = 0. \qquad (E-3)$$

Equation (E-3) provides a check on the computations for least squares adjustment.

APPENDIX F. THE CONTROVFRSIAL CRITERION FOR ASSIGNING WEIGHTS

1. The usual criterion for assigning weights is stated as:

a) the weights are inversely proportional to the squared standard deviations, i.e.,

$$\omega_i \, S_i^2 = K \qquad (F-1)$$

where K is an arbitrary constant;

b) the least precise observation has the unity weight, i.e.,

$$\omega_o S_o^2 = K \quad \Rightarrow \quad K = (1) \cdot S_o^2 \quad , \; or$$

$$K = S_o^2 \qquad (F-2)$$

where S_o is the standard deviation of least precise observation.

2. For the moment, consider only equation (F-1). The influence of the value assigned to k on the computations for obtaining the adjusted values and standard deviations of adjusted values will be determined. From eq (F-1) it will be obtained that

$$\omega_i = K / S_i^2 .$$

Then, the weight matrix W will be

$$W = \begin{bmatrix} K/S_1^2 & & & \\ & K/S_2^2 & & \\ & & \ddots & \\ & & & K/S_n^2 \end{bmatrix} = K \begin{bmatrix} 1/S_1^2 & & & \\ & 1/S_2^2 & & \\ & & \ddots & \\ & & & 1/S_n^2 \end{bmatrix} = K W'$$

and the trace $TR(W) = K/S_1^2 + \ldots + K/S_n^2 = K\,TR(W')$.

a) Computing adjusted values,

$$X = (A^T W A)^{-1} (A^T W L) =$$

$$= (1/K)(A^T W' A)^{-1} \cdot K (A^T W' L) = (A^T W' A)^{-1} (A^T W' L).$$

Therefore, it is concluded that, for the adjusted values X, the value assigned to K is, in fact, arbitrary.

b) Computing standard deviations of adjusted values,

$S_{x_i} = S_0 \sqrt{q_{ii}}$ where q_{ii} is an element of matrix Q.

Recalling that

$$Q = (A^T W A)^{-1} = (1/K)(A^T W' A)^{-1} = (1/K)\,Q',$$

then

$$q_{ii} = (1/K)\,q'_{ii}.$$

Next, by considering

$$S_0 = \sqrt{\frac{V^T W V}{TR(W) - m}} = \sqrt{K} \cdot \sqrt{\frac{V^T W' V}{K\,TR(W') - m}},$$

it follows that

$$S_{x_i} = \sqrt{q'_{ii}} \cdot \sqrt{\frac{V^T W' V}{K\,TR(W') - m}}.$$

Therefore, as should be expected, the standard deviations of adjusted values are affected by the value assigned to K.

3. In fact, eq (F-2) imposes a constraint on the K value:

$$K = S_o^2 .$$

To illustrate the consequences of accepting that kind of constraint, suppose that, given 100 observations, 99 are equally precise and one is less precise, say, with a standard deviation 400 times greater than the standard deviation of the remaining 99 observations. Then, according to the usual criterion, the least precise observation has the unity weight and the other 99 observations have the weight 20. That distribution of weights does not seem "good," and it is the author's opinion that there should be a better constraint minimizing the disturbances introduced by the assignment of different weights.

APPENDIX G. DECISION OF ADJUSTED VALUES

Given the observation equations

$$\begin{cases} a_1 x + b_1 y - l_1 = V_1 \\ a_2 x + b_2 y - l_2 = V_2 \\ a_3 x + b_3 y - l_3 = V_3 , \end{cases}$$

the standard deviations of adjusted values (by least squares method) for x and y will be determined [REF.2].

1. Assuming the observations were equally weighted, and solving the normal equations, the following is obtained:

$$\begin{cases} x = \dfrac{[ab][bl]-[b^2][al]}{[ab]^2-[a^2][b^2]} \\ \\ y = \dfrac{[ab][al]-[a^2][bl]}{[ab]^2-[a^2][b^2]} \end{cases} \qquad (G-1)$$

where the brackets have the usual meaning of sum.

Rearranging (G-1),

$$\begin{cases} x = \alpha_1 l_1 + \alpha_2 l_2 + \alpha_3 l_3 \\ y = \beta_1 l_1 + \beta_2 l_2 + \beta_3 l_3 \end{cases}$$

where

$$\alpha_i = \frac{[b^2]a_i - [ab]b_i}{[a^2][b^2]-[ab]^2}$$

$$\beta_i = \frac{[a^2]b_i - [ab]a_i}{[a^2][b^2]-[ab]^2} .$$

Consider l_1, l_2 and l_3 as values assumed, respectively, by independent random variables L_1, L_2 and L_3. Since the observations were equally weighted, then L_1, L_2 and L_3 present the same standard deviation, say, S_0. Then,

$$\begin{cases} VAR(X) = \sum_{i=1}^{3} \alpha_i^2 \, VAR(L_i) = [\alpha^2] S_0^2 \\ VAR(Y) = \sum_{i=1}^{3} \beta_i^2 \, VAR(L_i) = [\beta^2] S_0^2 \\ COVAR(X,Y) = \sum_{i=1}^{3} \alpha_i \beta_i \, VAR(L_i) = [\alpha\beta] S_0^2 . \end{cases} \qquad (G-2)$$

2. Determining the matrix $Q = (A^T W A)^{-1}$, the result is

$$Q = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} = \begin{bmatrix} \dfrac{[b^2]}{[a^2][b^2]-[ab]^2} & \dfrac{-[ab]}{[a^2][b^2]-[ab]^2} \\ \dfrac{-[ab]}{[a^2][b^2]-[ab]^2} & \dfrac{[a^2]}{[a^2][b^2]-[ab]^2} \end{bmatrix} .$$

3. Since

$$[\alpha^2] = \frac{[b^2]}{[a^2][b^2]-[ab]^2} = q_{11}$$

$$[\beta^2] = \frac{[a^2]}{[a^2][b^2]-[ab]^2} = q_{22}$$

$$[\alpha\beta] = \frac{-[ab]}{[a^2][b^2]-[ab]^2} = q_{12}$$

it may be concluded from eq. (G-2) that

$$\begin{cases} S_x = S_o \cdot \sqrt{q_{11}} \\ S_y = S_o \cdot \sqrt{q_{22}} \\ S_{xy} = S_o^2 \cdot q_{12} \; . \end{cases}$$

4. Finally, the correlation coefficient ρ between random variables x and y is given by

$$\rho = \frac{S_{xy}}{S_x \cdot S_y} = \frac{q_{12}}{[q_{11} \cdot q_{22}]^{1/2}} \; .$$

APPENDIX H. ERROR ELIPSE

1. The position $P(x\ y)$ of a vessel at sea is a two-dimensional random variable; its density function is the joint density function of the random variables x and y ;

$$f_{XY}(x\ y) = \frac{e^{-\left[\frac{\sigma_x^2\sigma_y^2}{2(\sigma_x^2\sigma_y^2-\sigma_{xy}^2)}\left[\frac{(x-\mu_x)^2}{\sigma_x^2} - 2\,\sigma_{xy}\,\frac{(x-\mu_x)(y-\mu_y)}{\sigma_x^2\,\sigma_y^2} + \frac{(y-\mu_y)^2}{\sigma_y^2}\right]\right]}}{2\pi\sqrt{\sigma_x^2\sigma_y^2-\sigma_{xy}^2}} .$$

Since μ_x and μ_y are the adjusted coordinates $(x_o y_o)$, if the origin of the coordinate system is positioned there, it will be obtained that

$$f_{XY}(x\ y) = \frac{e^{-\left[\frac{\sigma_x^2\sigma_y^2}{2(\sigma_x^2\sigma_y^2-\sigma_{xy}^2)}\left[\frac{x^2}{\sigma_x^2} - 2\,\sigma_{xy}\,\frac{x y}{\sigma_x^2\,\sigma_y^2} + \frac{y^2}{\sigma_y^2}\right]\right]}}{2\pi\sqrt{\sigma_x^2\sigma_y^2-\sigma_{xy}^2}} .$$

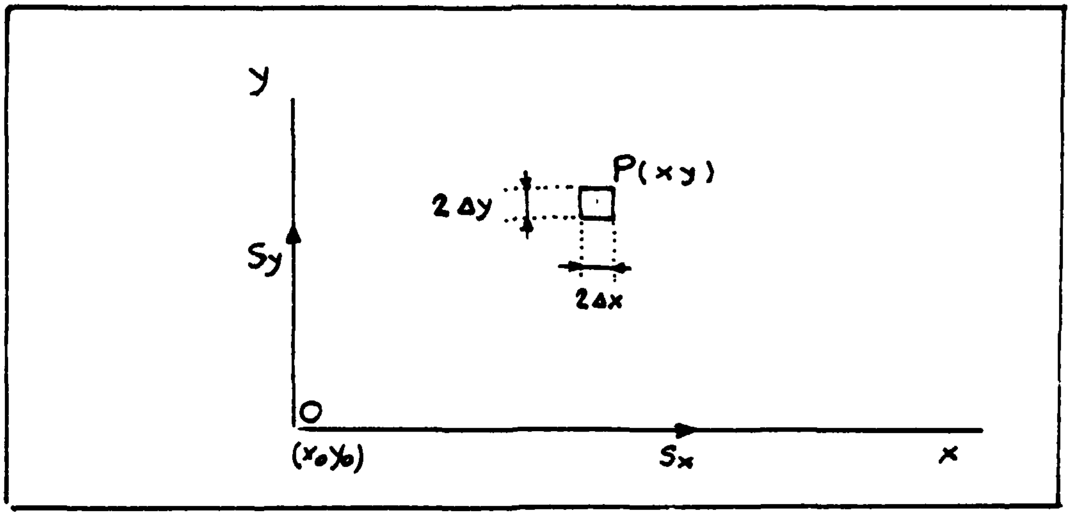


FIG H-1: TWO-DIMENSIONAL NORMAL DISTRIBUTION

Then, the probability of occurrence of the vessel's position in a small area ($2\Delta x \cdot 2\Delta y$) around $P(x,y)$ will be given by (see FIG H-1)

$$P(X = x \pm \Delta x,\ Y = y \pm \Delta y) = \int_{x-\Delta x}^{x+\Delta x} \int_{y-\Delta y}^{y+\Delta y} f_{XY}(x\,y)\, dy\, dx \,.$$

2. Now, it will be determined in what kind of line points with equal probability of occurrence are situated.

These points will present the same value f for the density function. Therefore, letting

$$K_1 = \sigma_x^2 \sigma_y^2 - \sigma_{xy}^2$$

$$K_2 = \left[- \ln (2\pi \sqrt{K_1}\, f)(2 K_1)\right] / \sigma_x^2 \sigma_y^2$$

and $K_3 = - K_2 \sigma_x^2 \sigma_y^2$, then

$$\sigma_y^2 x^2 - 2\sigma_{xy}\, x\, y + \sigma_x^2 y^2 + K_3 = 0 \qquad (H-1)$$

That quadratic equation in x and y represents a conic; the existence of the xy-term indicates that the conic is rotated out of its standard position.

a. Before determining what kind of conic equation (H-1) represents, check if points $(\sigma_x, 0)$ and $(0, \sigma_y)$ are both over the same contour line for a constant density function.

Inserting point $(\sigma_x, 0)$ into (H-1) results in $K_3 = -\sigma_x^2 \sigma_y^2$.

Inserting point $(0, \sigma_y)$ into (H-1) results in $K_3 = -\sigma_x^2 \sigma_y^2$.

Therefore, the points $(\sigma_x, 0)$ and $(0, \sigma_y)$ are over the same contour line (corresponding to $K_3 = -\sigma_x^2 \sigma_y^2$).

b. The analytical expression for the specific contour line containing points $(\sigma_x, 0)$ and $(0, \sigma_y)$ is

$$\sigma_y^2 x^2 - 2\,\sigma_{xy}\, x y + \sigma_x^2 y^2 - \sigma_x^2 \sigma_y^2 = 0 \qquad \text{(H-2)}$$

Considering

$$A = \sigma_y^2 \qquad C = \sigma_x^2 \qquad E = 0$$

$$B = -2\sigma_{xy} \qquad D = 0 \qquad F = -\sigma_x^2 \sigma_y^2$$

it is concluded that the discriminant is less than or equal to zero, i.e.,

$$B^2 - 4AC \leq 0 .$$

Therefore, if $B^2 - 4AC < 0$ then equation (H-2) represents an ellipse; if $B^2 - 4AC = 0$ then it will represent a straight line (a degenerate ellipse corresponding to a perfect correlation between random variables x and y).

3. The equation of the error ellipse in standard position:

Consider

$$(A^T W A)^{-1} = Q = \begin{bmatrix} q_1 & q_3 \\ q_3 & q_2 \end{bmatrix} .$$

Recalling that

$$S_x = S_o \cdot \sqrt{q_1}$$

$$S_y = S_o \cdot \sqrt{q_2}$$

and

$$S_{xy} = S_o^2 \cdot q_3$$

then equation (H-2) is equivalent to

$$q_2 x^2 - 2 q_3 xy + q_1 y^2 - q_1 q_2 S_o^2 = 0 . \qquad (H-3)$$

Consider

$$A = q_2 \qquad C = q_1 \qquad E = 0$$
$$B = -2q_3 \qquad D = 0 \qquad F = -q_1 q_2 S_o^2 .$$

If the x-axis is rotated an angle γ_o such that

$$\cot 2\gamma_o = \frac{A-C}{B} = \frac{q_1 - q_2}{2q_3} \qquad (H-4)$$

then the equation of the ellipse (H-3) in its standard position is

$$\frac{x^2}{\left(\dfrac{q_1 q_2 S_o^2}{q_2 \cos^2\gamma_o - 2q_3 \cos\gamma_o \sin\gamma_o + q_1 \sin^2\gamma_o}\right)} +$$

$$\frac{y^2}{\left(\dfrac{q_1 q_2 S_o^2}{q_2 \sin^2\gamma_o + 2q_3 \cos\gamma_o \sin\gamma_o + q_1 \cos^2\gamma_o}\right)} = 1 . \qquad (H-5)$$

After some algebraic and trigonometric manipulation, the following expression is obtained from (H-5) :

$$\frac{x^2}{\left(\dfrac{2q_1 q_2 S_o^2}{q_1 + q_2 - D}\right)} + \frac{y^2}{\left(\dfrac{2q_1 q_2 S_o^2}{q_1 + q_2 + D}\right)} = 1 \qquad (H-6)$$

where

$$D = \left[(q_1 - q_2)^2 + 4\, q_3^2 \right]^{1/2}. \qquad (H-7)$$

4. If the positive value of D satisfying eq. (H-7) is choosen, then the semi-major axis of the error ellipse is positioned along the "new" x-axis. Therefore, from the two solutions $\alpha_1 = \alpha$ and $\alpha_2 = \alpha + 90^\circ$ satisfying eq. (H-4) the valid one must be choosen (considering α as the smallest positive angle satisfying (H-4)).

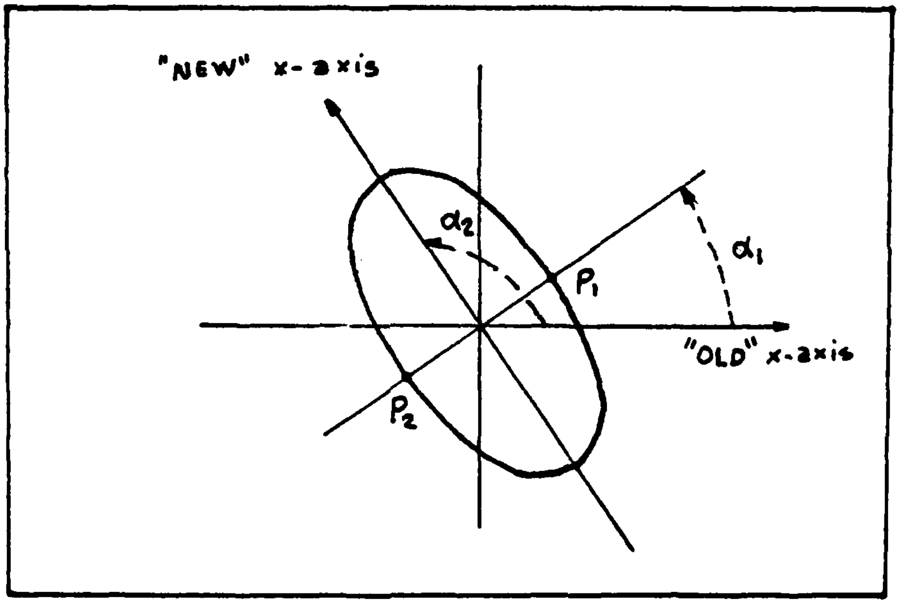


FIG H-2: ERROR ELLIPSE

The only way to solve that ambiguity is to test either with α_1 or α_2.

For that purpose, it is recommended to

a) obtain the point of intersection (either P_1 or P_2)

of the line $y = x \tan\alpha$ with the ellipse before rotation (expressed by eq (H-3));

b) determine the distance $\underline{d}$ between the origin and P_1 (or P_2);

c) if $d \doteq$ semi-major axis a, then the major axis makes an angle $\gamma_0 = \alpha_1$ (measured counterclockwise) with the "old" x-axis; if not, then the angle will be $\gamma_0 = \alpha_2 = \alpha_1 + 90^\circ$, that is, the major axis runs through 2nd and 4th quadrant.

APPENDIX I - ALGORITHMS

```
MODULE 1
ALGORITHM FIX-BY-N-AZIMUTHS
   INPUT N
   PI←3.141592653589793
   OUTPUT'NUMBER OF STATIONS=',N
   DO FOR I←1 TO N
      INPUT TABLE-INPUT(I,1),TABLE-INPUT(I,2),TABLE-INPUT(I,4)
      OUTPUT'ST#',I,'EAST=',TABLE-INPUT(I,1),'NORT=',
         TABLE-INPUT(I,2),'ST ERROR=',TABLE-INPUT(I,4)
   END DO
   DO FOR I←1 TO N
      INPUT TABLE-INPUT(I,3)
   END DO
   DO WHILE TABLE-INPUT(1,3)≠400.0
      OUTPUT'OBSERVED AZIMUTHS'
      DO FOR I←1 TO N
         OUTPUT'AZIMUTH FROM STATION#',I,'=',TABLE-INPUT(I,3),
            'DEGREES'
         ALGORITHM CONVERSION-DEGREES-RADIANS(TABLE-INPUT(I,3))
            TABLE-INPUT(I,3)←TABLE-INPUT(I,3)*(PI/180.0)
         END CONVERSION-DEGREES-RADIANS(TABLE-INPUT(I,3))
      END DO
      MODULES 2,3,4,5,6,7
      DO FOR I←1 TO N
         INPUT TABLE-INPUT(I,3)
      END DO
   END DO
END FIX-BY-N-AZIMUTHS

MODULE 2
ALGORITHM WEIGHT-MATRIX (N,TABLE-INPUT(I,4))
   ALGORITHM ZERO (TABLE-WEIGHT)
      DO FOR I←1 TO 10
         DO FOR J←1 TO 10
            TABLE-WEIGHT(I,J)←0.000
         END DO
      END DO
   END ZERO (TABLE-WEIGHT)
   ALGORITHM SQUARE (N,TABLE-INPUT(I,4),TABLE-WEIGHT)
      DO FOR I←1 TO N
         TABLE-WEIGHT(I,I)←TABLE-INPUT(I,4)**2
      END DO
   END SQUARE (TABLE-WEIGHT)
   ALGORITHM NORMALIZE (TABLE-WEIGHT)
      GREATEST←TABLE-WEIGHT(1,1)
      DO FOR I←2 TO N
         IF TABLE-WEIGHT(I,I) > GREATEST THEN
            GREATEST←TABLE-WEIGHT(I,I)
         END IF
         END DO
```

```
      DO FOR I←1 TO N
         TABLE-WEIGHT(I,I)←GREATEST/TABLE-WEIGHT(I,I)
      END DO
   END NORMALIZE (TABLE-WEIGHT)
END WEIGHT-MATRIX (TABLE-WEIGHT)

MODULE 3
ALGORITHM FIRST-INITIAL-POINT (TABLE-INPUT,N)
   ALGORITHM SELECT-AZIMUTHS(TABLE-INPUT(I,3),N)
      I←2
      DO WHILE TANGENT(TABLE-INPUT(I,3))=TANGENT(TABLE-
         INPUT(1,3))
         I←I+1
         IF I>N THEN
            OUTPUT'POSITION IS UNDETERMINED FOR THAT DATA SET'
            PICK UP ANOTHER DATA SET
         END IF
      END DO
   END SELECT-AZIMUTH(I)
   ALGORITHM INITIAL-COORDINATES (TABLE-INPUT,I)
      IF TABLE-INPUT(1,3)=0.0 OR TABLE-INPUT(1,3)=PI THEN
         MK←TANGENT((5./2.)*PI-TABLE-INPUT(I,3))
         XO←TABLE-INPUT(1,1)
         YO←TABLE-INPUT(I,2)+MK*(XO-TABLE-INPUT(I,1))
      ELSE IF TABLE-INPUT(I,3)=0.0 OR TABLE-INPUT(I,3)=PI THEN
         MI←TANGENT((5./2.)*PI-TABLE-INPUT(1,3))
         XO←TABLE-INPUT(I,1)
         YO←TABLE-INPUT(1,2)+MI*(XO-TABLE-INPUT(1,1))
      ELSE
         MI←TANGENT((5./2.)*PI-TABLE-INPUT(1,3))
         MK←TANGENT((5./2.)*PI-TABLE-INPUT(I,3))
         XO←(TABLE-INPUT(I,2)-TABLE-INPUT(1,2)+MI*TABLE-
            INPUT(1,1)-MK*TABLE-INPUT(I,1))/(MI-MK)
         YO←TABLE-INPUT(1,2)+MI*(XO-TABLE-INPUT(1,1))
      END IF
   END INITIAL-COORDINATES (XO,YO)
END FIRST-INITIAL-POINT(XO,YO)

MODULE 4
ALGORITHM ITERATIONS (TOLERANCE)
   DO UNTIL TOLERANCE<1.0
      MODULES 8,9,10,11,12,13,14
   END DO
END ITERATIONS(XO,YO)

MODULE 5
ALGORITHM FINAL-ADJUSTED-POSITION (XO,YO)
   OUTPUT'ADJUSTED COORDINATES X=',XO,'Y=',YO
END FINAL-ADJUSTED-POSITION (X,Y)
```

```
MODULE 6
ALGORITHM PRECISION (TABLE-A,TABLE-WEIGHT,TABLE-Q,LIST-L,
   DELTAX,DELTAY,N)
   MODULES 15,16,17,18,19
END PRECISION(SU,SX,SY,SXY,RO)

MODULE 7
ALGORITHM ERROR-ELIPSE(TABLE-Q,SU)
   OUTPUT'ERROR ELIPSE SEMI-AXIS AND ORIENTATION'
   MODULES 20, 21, 22, 23, 24, 25, 26
END ERRO-ELIPSE(SA,SB,GAMAO)

MODULE 8
ALGORITHM INITIAL-AZIMUTHS(XO,YO,TABLE-INPUT,N)
   DO FOR I←1 TO N
      IF YO=TABLE-INPUT(I,2) AND XO > TABLE-INPUT(I,1) THEN
         LIST-AO(I)←PI/2.
      ELSE IF YO=TABLE-INPUT(I,2) AND XO < TABLE-INPUT(I,1) THEN
         LIST-AO(I)←(3.0*PI)/2.
      ELSE IF XO=TABLE-INPUT(I,1) AND YO >TABLE-INPUT(I,2) THEN
         LIST-AO(I)←0.0
      ELSE IF XO=TABLE-INPUT(I,1) AND YO <TABLE-INPUT(I,2) THEN
         LIST-AO(I)← PI
      ELSE
         ALFA(I)←ARC TANGENT((XO-TABLE-INPUT(I,1))/(YO-
            TABLE-INPUT(I,2)))
         IF ALFA(I) >0.0 AND XO >TABLE-INPUT(I,1) THEN
            LIST-AO(I)← ALFA(I)
      ELSE IF ALFA(I) >0.0 AND XO < TABLE-INPUT(I,1) THEN
         LIST-AO(I)← ALFA(I)+PI
      ELSE IF ALFA(I) < 0.0 AND XO >TABLE-INPUT(I,1) THEN
         LIST-AO(I)← ALFA(I)+PI
      ELSE
         LIST-AO(I)←ALFA(I)+2.0*PI
         END IF
      END IF
   END DO
END INITIAL-AZIMUTHS(LIST-AO)

MODULE 9
ALGORITHM MATRIX-L(TABLE-INPUT(I,3),LIST-AO,N)
   ALGORITHM ZERO(LIST-L)
      DO FOR I←1 TO 10
         LIST-L(I)←0.000
      END DO
   END ZERO(LIST-L)
   DO FOR I←1 TO N
      LIST-L(I)←TABLE-INPUT(I,3)-LIST-AO(I)
   END DO
END MATRIX-L(LIST-L)
```

```
MODULE 10
ALGORITHM INITIAL-SQUARED-DISTANCES(XO,YO,N,TABLE-INPUT)
   DO FOR I←1 TO N
      LIST-SO(I)←(XO-TABLE-INPUT(I,1))**2+(YO-TABLE-
         INPUT(I,2))**2
   END DO
END INITIAL-SQUARED-DISTANCES(LIST-SO)

MODULE 11
ALGORITHM MATRIX-A (N,TABLE-INPUT,XO,YO,LIST-SO)
   ALGORITHM ZERO (TABLE-A)
      DO FOR I←1 TO 10
         DO FOR J←1 TO 2
            TABLE-A(I,J)←0.000
         END DO
      END DO
   END ZERO(TABLE-A)
   ALGORITHM ELEMENTS(TABLE-A,N,TABLE-INPUT,XO,YO,LIST-SO)
      DO FOR I←1 TO N
         TABLE-A(I,1)←(YO-TABLE-INPUT(I,2))/LIST-SO(I)
      END DO
      DO FOR I←1 to N
         TABLE-A(I,2)←(TABLE-INPUT(I,1)-XO)/LIST-SO(I)
      END DO
   END ELEMENTS(TABLE-A)
END MATRIX-A(TABLE-A)

MODULE 12
ALGORITHM NORMAL-EQUATIONS (TABLE-A,TABLE-WEIGHT,LIST-L)
   ALGORITHM ATW(TABLE-A,TABLE-WEIGHT)
      DO FOR I←1 TO 2
         DO FOR J←1 TO 10
            TABLE-ATW(I,J)←0.00
            DO FOR K←1 TO 10
               TABLE-ATW(I,J)←TABLE-ATW(I,J)+TABLE-
               A(K,I)*TABLE-WEIGHT(K,J)
            END DO
         END DO
      END DO
   END ATW(TABLE-ATW)
   ALGORITHM ATWA (TABLE-ATW,TABLE-A)
      DO FOR I←1 TO 2
         DO FOR J←1 TO 2
            TABLE-ATWA(I,J)←0.00
            DO FOR K←1 TO 10
               TABLE-ATWA(I,J)←TABLE-ATWA(I,J)+TABLE-
               ATW(I,K)*TABLE-A(K,J)
            END DO
         END DO
      END DO
   END ATWA (TABLE-ATWA)
```

```
   ALGORITHM INVERT-ATWA(TABLE-ATWA)
      BETA←TABLE-ATWA(1,2)**2-TABLE-ATWA(1,1)*TABLE-ATWA(2,2)
      TABLE-Q(1,1)← -TABLE-ATWA(2,2)/BETA
      TABLE-Q(1,2)←TABLE-ATWA(1,2)/BETA
      TABLE-Q(2,1)←TABLE-Q(1,2)
      TABLE-Q(2,2)←-TABLE-ATWA(1,1)/BETA
   END INVERT-ATWA(TABLE-Q)
   ALGORITHM ATWL(TABLE-ATW,LIST-L)
      DO FOR I←1 TO 2
         LIST-ATWL(I)←0.0
         DO FOR K←1 TO 10
            LIST-ATWL(I)←LIST-ATWL(I)+TABLE-ATW(I,K)*LIST-L(K)
         END DO
      END DO
   END ATWL(LIST-ATWL)
   ALGORITHM ADJUSTED-INCREMENTS(TABLE-Q,LIST-ATWL)
      DELTAX←TABLE-Q(1,1)*LIST-ATWL(1)+TABLE-
      Q(1,2)*LIST-ATWL(2)
      DELTAY←TABLE-Q(2,1)*LIST-ATWL(1)+TABLE-
      Q(2,2)*LIST-ATWL(2)
   END ADJUSTED-INCREMENTS(DELTAX,DELTA Y)
END NORMAL-EQUATIONS(DELTAX,DELTA Y)

MODULE 13
ALGORITHM NEW-INITIAL-POINT (XO,YO,DELTAX,DELTAY)
   XO←XO+DELTAX
   YO←YO+DELTAY
END NEW-INITIAL-POINT(XO,YO)

MODULE 14
ALGORITHM TOLERANCE(DELTAX,DELTAY)
   TOLERANCE←DELTAX**2+DELTAY**2
END TOLERANCE(TOLERANCE)

MODULE 15
ALGORITHM RESIDUALS(TABLE-A,LIST-L,DELTAX,DELTAY,N)
   LIST-X(1)←DELTAX
   LIST-X(2)←DELTAY
   ALGORITHM LIST-AX(TABLE-A,LIST-X,N)
      DO FOR I←1 TO N
         LIST-AX(I)←0.0
         DO FOR J←1 TO 2
            LIST-AX(I)←LIST-AX(I)+TABLE-A(I,J)*LIST-X(J)
         END DO
      END DO
   END LIST-AX(LIST-AX)
```

```
      ALGORITHM LIST-V(LIST-AX,LIST-L,N)
         DO FOR I←1 TO N
            LIST-V(I)←LIST-AX(I)-LIST-L(I)
         END DO
      END LIST-V(LIST-V)
   END RESIDUALS(LIST-V)

   MODULE 16
   ALGORITHM ST-DEVIATION-OF-UNIT-WEIGHT-OBS(LIST-V,TABLE-WEIGHT,N)
      ALGORITHM LIST-VTW(LIST-V,TABLE-WEIGHT,N)
         DO FOR I←1 TO N
            LIST-VTW(I)←LIST-V(I)*TABLE-WEIGHT(I,I)
         END DO
      END LIST-VTW(LIST-VTW)
      ALGORITHM VTWV(LIST-VTW,LIST-V)
         VTWV←0.0
         DO FOR I←1 TO N
            VTWV←VTWV+LIST-VTW(I)*LIST-V(I)
         END DO
      END VTWV(VTWV)
      ALGORITHM TRACE(TABLE-WEIGHT)
         TRACE←0.0
            DO FOR I←1 TO N
               TRACE←TRACE+TABLE-WEIGHT(I,I)
            END DO
      END TRACE(TRACE)
      ALGORITHM SU(VTWV,TRACE)
         CHARLIE←(VTWV/(TRACE-2.0))
         SU←SQRT(CHARLIE)
      END SU(SU)
   END ST-DEVIATION-OF-UNIT-WEIGHT-OBS(SU)

   MODULE 17
   ALGORITHM ST-DEVIATION-OF-EACH-OBS(SU,TABLE-WEIGHT)
      OUTPUT'PRECISION OF OBSERVATIONS'
      DO FOR I←1 TO N
         S←(SU/SQRT(TABLE-WEIGHT(I,I)))*(180.0/PI)
         OUTPUT'ST DEVIATION OF OBS',I,'=',S,'DEGREES'
      END DO
   END ST-DEVIATION-OF-EACH-OBS

   MODULE 18
   ALGORITHM ST-DEVIATIONS-AND-COVARIANCE-OF X-AND-Y(SU,TABLE-Q)
      SX←SU*SQRT(TABLE-Q(1,1))
      SY←SU*SQRT(TABLE-Q(2,2))
      SXY←(SU**2)*TABLE-Q(1,2)
      OUTPUT'SX=',SX,'SY=',SY,'SXY=',SXY
   END ST-DEVIATIONS-AND-COVARIANCE-OF-X-AND-Y(SX,SY,SXY)
```

```
MODULE 20
ALGORITHM D(TABLE-Q)
   D←SQRT((TABLE-Q(1,1)-TABLE-Q(2,2))**2+4.0*(TABLE-
      Q(1,2)**2))
END D(D)

MODULE 21
ALGORITHM SEMI-MAJOR-AXIS(SU,TABLE-Q)
   SA← SU*SQRT(2.0*TABLE-Q(1,1)*TABLE-Q(2,2)/(TABLE-Q(1,1)+
      TABLE-Q(2,2)-D))
   OUTPUT'SEMI-MAJOR AXIS SA=',SA
END SEMI-MAJOR-AXIS(SA)

MODULE 22
ALGORITHM SEMI-MINOR-AXIS(SU,TABLE-Q)
   SB ←SU*SQRT(2.0*TABLE-Q(1,1)*TABLE-Q(2,2)/(TABLE-Q(1,1)+
      TABLE-Q(2,2)+D))
   OUTPUT'SEMI-MINOR AXIS SB=',SB
END SEMI-MINOR-AXIS(SB)

MODULE 23
ALGORITHM GAMA(TABLE-Q)
   IF TABLE-Q(1,1)=TABLE-Q(2,2) THEN
      GAMA ←PI/4.0
   ELSE
      OMEGA ←ARC TANGENT(2.0*TABLE-Q(1,2)/(TABLE-Q(1,1)-
         TABLE-Q(2,2)))
      IF OMEGA > 0.0 THEN
      GAMA←OMEGA/2.0
   ELSE
      GAMA ←(OMEGA+PI)/2.0
      END IF
   END IF
END GAMA(GAMA)

MODULE 24
ALGORITHM INTERSECTION(SU,TABLE-Q,GAMA)
   X10 ←(SU**2)*TABLE-Q(1,1)*TABLE-Q(2,2)
   X11←TABLE-Q(2,2)-2.0*TABLE-Q(1,2)*TANGENT(GAMA)+
      (TANGENT(GAMA**2)*TABLE-Q(1,1)
   X1←X10/X11
   Y1←X1*(TANGENT(GAMA)**2)
END INTERSECTION (X1,Y1)

MODULE 25
ALGORITHM AVERAGE(SA,SB)
   AVER ←((SA+SB)/2.0)**2
END AVERAGE(AVER)
```

```
MODULE 26
ALGORITHM SELECTION(AVER,X1,Y1)
   D1 ← X1+Y1
   IF D1 > AVER THEN
      GAMAO ← GAMA
   ELSE
      GAMAO ←GAMA+PI/2.0
   END IF
   GAMAO ←GAMAO*(180.0/PI)
   OUTPUT'ANGLE FROM X-AXIS TO SA ANTICLOCKWISE=',GAMAO
END SELECTION(GAMAO)

MODULE 19
ALGORITHM CORRELATION-COEFFICIENT(SX,SY,SXY)
   RO ← SXY/(SX*SY)
   OUTPUT'CORRELATION COEFFICIENT RO=',RO
END CORRELATION-COEFFICIENT(RO)
```

```
MODULE 30
ALGORITHM FIX-BY-N-SEXTANT-ANGLES
   PI ← 3.141592653589793
   INPUT N
   OUTPUT'NUMBER OF SEXTANT ANGLES=',N
   M ← N+1
   DO FOR I ← 1 TO N
      INPUT TABLE-INPUT(I,1),TABLE-INPUT(I,2),TABLE-
         INPUT(I, 4)
      OUTPUT'ST#',I,'EAST=',TABLE-INPUT(I,1),'NORTH=',TABLE-
         INPUT(I,2),'ST ERROR=',TABLE-INPUT(I,4)
   END DO
   INPUT TABLE-INPUT(M,1),TABLE-INPUT(M,2)
   OUTPUT'ST#',M,'EAST=',TABLE-INPUT(M,1),'NORTH=',TABLE-
      INPUT(M,2)
   DO FOR I← 1 TO N
      INPUT TABLE-INPUT(I,3)
   END DO
   DO WHILE TABLE-INPUT(1,3)≠400.0
      OUTPUT'OBSERVED SEXTANT ANGLES'
      DO FOR I← 1 TO N
         J← I+1
         OUTPUT'SEXTANT ANGLE BETWEEN ST#',I,'AND ST#',J,
            '=',TABLE-INPUT(I,3),'DEGREES'
         ALGORITHM CONVERSION-DEGREES-RADIANS(TABLE-INPUT(I,3))
            TABLE-INPUT(I,3)  TABLE-INPUT(I,3)*(PI/180.0)
         END CONVERSION-DEGREES-RADIANS(TABLE-INPUT(I,3))
      END DO
      MODULES 2,31,32,5,6,7
      DO FOR I← 1 TO N
         INPUT TABLE-INPUT(I,3)
      END DO
   END DO
END FIX-BY-N-SEXTANT-ANGLES

MODULE 31
ALGORITHM FIRST-INITIAL-POINT-FOR-FIX-BY-N-SEXTANT-
   ANGLES(TABLE-INPUT)
   MODULES 33,34,35,36
END FIRST-INITIAL-POINT-FOR-FIX-BY-N-SEXTANT-ANGLES(XO.YO)

MODULE 32
ALGORITHM ITERATIONS(TOLERANCE)
   DO UNTIL TOLERANCE <1.000
      MODULES 37,38,39,40,12,13,14
   END DO
END ITERATIONS(XO,YO)
```

```
MODULE 33
ALGORITHM SELECT-SEXTANT-ANGLES(TABLE-INPUT)
   J←1
   DO UNTIL FRAC 1 ≠ FRAC 2
      J←J+1
      IF J ≥ M THEN
         OUTPUT'SOLUTION UNDETERMINED FOR THAT DATA SET'
         PICK UP ANOTHER DATA SET
      END IF
      I←J-1
      K←J+1
   ANGUL←TABLE-INPUT(I,3)+TABLE-INPUT(J,3)
   FRAC 1←COSINE(ANGUL)*SQRT(((TABLE-INPUT(I,1)-
      TABLE-INPUT(J,1))**2+(TABLE-INPUT(I,2)-TABLE-
      INPUT(J,2))**2)*((TABLE-INPUT(K,1)-TABLE-INPUT(J,1))**2+
      (TABLE-INPUT(K,2)-TABLE-INPUT(J,2))**2))
   FRAC 2←(TABLE-INPUT(I,1)-TABLE-INPUT(J,1))*(TABLE-
      INPUT(J,1)-TABLE-INPUT(K,1))+(TABLE-INPUT(I,2)
      -TABLE-INPUT(J,2))*(TABLE-INPUT(J,2)-TABLE-INPUT(K,2))
   END DO
END SELECT-SEXTANT-ANGLES(I,J,K)

MODULE 34
ALGORITHM INTERCHANGE-DATA(TABLE-INPUT,I,J,K)
   STORE(1)←TABLE-INPUT(1,1)
   STORE(2)←TABLE-INPUT(1,2)
   STORE(3)←TABLE-INPUT(1,3)
   STORE(4)←TABLE-INPUT(2,1)
   STORE(5)←TABLE-INPUT(2,2)
   STORE(6)←TABLE-INPUT(2,3)
   STORE(7)←TABLE-INPUT(3,1)
   STORE(8)←TABLE-INPUT(3,2)
   STORE(9)←TABLE-INPUT(I,1)
   STORE(10)←TABLE-INPUT(I,2)
   STORE(11)←TABLE-INPUT(I,3)
   STORE(12)←TABLE-INPUT(J,1)
   STORE(13)←TABLE-INPUT(J,2)
   STORE(14)←TABLE-INPUT(J,3)
   STORE(15)←TABLE-INPUT(K,1)
   STORE(16)←TABLE-INPUT(K,2)
   TABLE-INPUT(1,1)←STORE(9)
   TABLE-INPUT(1,2)←STORE(10)
   TABLE-INPUT(1,3)←STORE(11)
   TABLE-INPUT(2,1)←STORE(12)
   TABLE-INPUT(2,2)←STORE(13)
   TABLE-INPUT(2,3)←STORE(14)
   TABLE-INPUT(3,1)←STORE(15)
   TABLE-INPUT(3,2)←STORE(16)
END INTERCHANGE-DATA(TABLE-INPUT,STORE)
```

```
MODULE 35
ALGORITHM INITIAL-COORDINATES(TABLE-INPUT)
   IF TABLE-INPUT (1,3)≠(PI/2.0) AND TABLE-INPUT(2,3)≠(PI/(2.0)
      THEN
      AB ←TANGENT(TABLE-INPUT(1,3))
      BA← TANGENT(TABLE-INPUT(2,3))
      E ←(TABLE-INPUT(2,1)-TABLE-INPUT(1,1))/AB+(TABLE-
          INPUT(2,1)-TABLE-INPUT(3,1))/BA+TABLE-INPUT(3,2)-
          TABLE,INPUT(1,2)
      F ← (TABLE-INPUT(2,2))-TABLE-INPUT(1,2))/AB+(TABLE-INPUT(2.2)-
          TABLE-INPUT(3,2))/BA+TABLE-INPUT(1,1)-
          TABLE-INPUT(3,1)
      G ←(TABLE-INPUT(1,2)*TABLE-INPUT(2,1)-TABLE-INPUT(2,2)*
          TABLE-INPUT(1,1))/AB+(TABLE-INPUT(2,1)*TABLE-INPUT(3,2)-
          TABLE-INPUT(3,1)*TABLE-INPUT(2,2))/BA+TABLE-
          INPUT(2,1)*TABLE-INPUT(3,1)+TABLE-INPUT(2,2)*TABLE-
          INPUT(3,2)-TABLE-INPUT(1,1)*TABLE-INPUT(2,1)-
          TABLE-INPUT(1,2)*TABLE-INPUT(2,2)
   IF F=0.0 THEN
      DAO←G/E
      YO1←DAO
      YO2←DAO
      U←AB
      R←TABLE-INPUT(1,2)-TABLE-INPUT(2,2)-AB*(TABLE-INPUT(1,1)+
         TABLE-INPUT(2,1))
      SAL← AB*(DAO**2-DAO*(TABLE-INPUT(1,2)+TABLE-INPUT(2,2))+
         TABLE-INPUT(1,1)*TABLE-INPUT(2,1)+TABLE-INPUT(1,2)*
         TABLE-INPUT(2,2))+DAO*(TABLE-INPUT(2,1)-TABLE-INPUT(1,1))+
         TABLE-INPUT(1,1)*TABLE-INPUT(2,2)-TABLE-INPUT(2,1)*
         TABLE-INPUT(1,2)
      DISC←SQRT(R**2-4.0*U*SAL)
      XO1 ← (-R+DISC)/(2.0*U)
      XO2 ← (-R-DISC)/(2.0*U)
   ELSE IF E=0.0 THEN
      H←(-G/F)
      XO1←H
      XO2←H
      U ← AB
      R ← TABLE-INPUT(2,1)-TABLE-INPUT(1,1)-AB*(TABLE-INPUT(1,2)+
          TABLE-INPUT(2,2))
      SAL←AB*(H**2-H*(TABLE-INPUT(1,1)+TABLE-INPUT(2,1))+
          TABLE-INPUT(1,1)*TABLE-INPUT(2,1)+TABLE-INPUT(1,2)*
          TABLE-INPUT(2,2))+H*(TABLE-INPUT(1,2)-TABLE-INPUT(2,2))+
          TABLE-INPUT(1,1)*TABLE-INPUT(2,2)-TABLE-INPUT(2,1)*
          TABLE-INPUT(1,2)
      DISC←SQRT(R**2-4.0*U*SAL)
      YO1 ← (-R+DISC)/(2.0*U)
      YO2 ← (-R-DISC)/(2.0*U)
   ELSE
      C←F/E
```

```
    DAO ← G/E
    U  ← AB*(C**2+1.0)
    R  ← AB*(2.0*C*DAO-C*(TABLE-INPUT(1,2)+TABLE-INPUT(2,2))
         -TABLE-INPUT(1,1)-TABLE-INPUT(2,1))+C*(TABLE-
         INPUT(2,1)-TABLE-INPUT(1,1))+TABLE-INPUT(1,2)-
         TABLE-INPUT(2,2)
    SAL ← AB*(DAO**2-DAO*(TABLE-INPUT(1,2)+TABLE-INPUT(2,2))+
         TABLE-INPUT(1,1)*TABLE-INPUT(2,1)+TABLE-INPUT(1,2)*
         TABLE-INPUT(2,2))+DAO*(TABLE-INPUT(2,1)-TABLE-
         INPUT(1,1))+TABLE-INPUT(1,1)*TABLE-INPUT(2,2)-
         TABLE-INPUT(2,1)*TABLE-INPUT(1,2)
    DISC ← SQRT(R**2-4.0*U*SAL)
    X01 ← (-R+DISC)/(2.0*U)
    X02 ← (-R-DISC)/(2.0*U)
    Y01 ← C*X01+DAO
    Y02 ← C*X02+DAO
END IF
ELSE IF TABLE-INPUT(1,3)=(PI/2.0) AND TABLE-INPUT(2,3)≠
    (PI/2.0) THEN
    BA ← TANGENT(TABLE-INPUT(2,3))
    E ← BA*(TABLE-INPUT(3,2)-TABLE-INPUT(1,2))+
            TABLE-INPUT(2,1)-TABLE-INPUT(3,1)
    F ← BA*(TABLE-INPUT(1,1)-TABLE-INPUT(3,1))+TABLE-INPUT(2,2)-
            TABLE-INPUT(3,2)
    G ← BA*(TABLE-INPUT(2,2)*TABLE-INPUT(3,2)+TABLE-INPUT(2,1)*
            TABLE-INPUT(3,1)-TABLE-INPUT(1,2)*TABLE-INPUT(2,2)-
            TABLE-INPUT(1,1)*TABLE-INPUT(2,1))+
            TABLE-INPUT(2,1)*TABLE-INPUT(3,2)-TABLE-INPUT(3,1)*
            TABLE-INPUT(2,2)
    IF F=0.0 THEN
       DAO ← G/E
       Y01 ← DAO
       Y02 ← DAO
       R  ← -TABLE-INPUT(1,1)-TABLE-INPUT(2,1)
       SAL ← DAO**2-DAO*(TABLE-INPUT(1,2)+TABLE-INPUT(2,2))+
             TABLE-INPUT(1,2)*TABLE-INPUT(2,2)+TABLE-
             INPUT(1,1)*TABLE-INPUT(2,1)
       DISC ← SQRT(R**2-4.0*SAL)
       X01 ← (-R+DISC)/2.0
       X02 ← (-R-DISC)/2.0
    ELSE IF E=0.0 THEN
       H ← (-G/F)
       X01 ← H
       X02 ← H
       R ← -TABLE-INPUT(1,2)-TABLE-INPUT(2,2)
       SAL ← H**2-H*(TABLE-INPUT(1.1)+TABLE-INPUT(2.1)+
           TABLE-INPUT(1,1))*TABLE-INPUT(2,1)+TABLE-
           INPUT(1,2)*TABLE-INPUT(2,2)
       DISC ← SQRT(R**2-4.0*SAL)
       Y01 ← (-R+DISC)/2.0
       Y02 ← (-R-DISC)/2.0
```

```
ELSE
   C←F/E
   DAO ←G/E
   U←C**2+1.0
   R←2.0*C*DAO-C*(TABLE-INPUT(1,2)+TABLE-INPUT(2,2))
      -TABLE-INPUT(1,1)-TABLE-INPUT(2,1)
   SAL←DAO**2-DAO*(TABLE-INPUT(1,2)+TABLE-INPUT(2,2))+
      TABLE-INPUT(1,2)*TABLE-INPUT(2,2)+TABLE-INPUT(1,1)*
      TABLE-INPUT(2,1)
   DISC←SORT(R**2-4.0*U*SAL)
   X01←(-R+DISC)/(2.0*U)
   X02←(-R-DISC)/(2.0*U)
   Y01←C*X01+DAO
   Y02←C*X02+DAO
END IF
ELSE IF TABLE-INPUT(1,3)≠(PI/2.0)AND TABLE-INPUT(2,3)=
   (PI/2.0) THEN
   AB←TANGENT(TABLE-INPUT(1,3))
   E ← AB*(TABLE-INPUT(1,2)-TABLE-INPUT(3,2))+TABLE-
      INPUT(1,1)-TABLE-INPUT(3,1)
   F←AB*(TABLE-INPUT(3,1)-TABLE-INPUT(1,1))+TABLE-
      INPUT(1,2)-TABLE-INPUT(2,2)
   G ←AB*(TABLE-INPUT(1,1)*TABLE-INPUT(2,1)+TABLE-INPUT(1,2)*
      TABLE-INPUT(2,2)-TABLE-INPUT(2,1)*TABLE-INPUT(3,1)-
      TABLE-INPUT(2,2)*TABLE-INPUT(3,2))+TABLE-INPUT(1,1)*
      TABLE-INPUT(2,2)-TABLE-INPUT(2,1)*TABLE-INPUT(1,2)
   IF F=0.0 THEN
      DAO←G/E
      Y01←DAO
      Y02←DAO
      R←-TABLE-INPUT(2,1)-TABLE-INPUT(3,1)
      SAL ←DAO**2-DAO*(TABLE-INPUT(2,2)+TABLE-INPUT(3,2))+
         TABLE-INPUT(2,2)*TABLE-INPUT(3,2)+TABLE-INPUT(2,1)*
         TABLE-INPUT(3,1)
      DISC←SQRT(R**2-4.0*SAL)
      X01←(-R+DISC)/2.0
      X02 ←(-R-DISC)/2.0
ELSE IF E=0.0 THEN
   H←(-G/F)
   X01←H
   X02←H
   R← -TABLE-INPUT(2,2)-TABLE-INPUT(3,2)
   SAL← H**2-H*(TABLE-INPUT(2,1)+TABLE-INPUT(3,1))+
       TABLE-INPUT(2,1)*TABLE-INPUT(3,1)+TABLE-INPUT(2,2)*
       TABLE-INPUT(3,2)
   DISC←SQRT(R**2-4.0*SAL)
   Y01←(-R+DISC)/2.0
   Y02←(-R-DISC)/2.0
ELSE
   C←F/E
   DAO←G/E
```

```
      U ← C**2+1.0
      R ← 2.0*C*DAO-C*(TABLE-INPUT(2,2)+TABLE-INPUT(3,2))-
         TABLE-INPUT(2,1)-TABLE-INPUT(3,1)
      SAL ← DAO**2-DAO*(TABLE-INPUT(2,2)+TABLE-INPUT(3,2))+TABLE-
         INPUT(2,1)*TABLE-INPUT(3,1)+TABLE-INPUT(2,2)*TABLE-
         INPUT(3,2)
      DISC ← SQRT(R**2-4.0*U*SAL)
      X01 ← (-R+DISC)/(2.0*U)
      X02 ← (-R-DISC)/(2.0*U)
      Y01 ← (C*X01+DAO
      Y02 ← C*X02+DAO
      END IF
   ELSE
      E ← TABLE-INPUT(1,2)-TABLE-INPUT(3,2)
      F ← TABLE-INPUT(3,1)-TABLE-INPUT(1,1)
      G ← TABLE-INPUT(1,1)*TABLE-INPUT(2,1)+TABLE-
         INPUT(1,2)*TABLE-INPUT(2,2)-TABLE-INPUT(2,2)*TABLE-
         INPUT(3,2)-TABLE-INPUT(2,1)*TABLE-INPUT(3,1)
      IF F=0.0 THEN
         DAO ← G/E
         Y01 ← DAO
         Y02 ← DAO
         X01 ← TABLE-INPUT(1,1)
         X02 ← TABLE-INPUT(1,1)
      ELSE IF E=0.0 THEN
         H ← (-G/F)
         X01 ← H
         X02 ← H
         Y01 ← TABLE-INPUT(1,2)
         Y02 ← TABLE-INPUT(1,2)
      ELSE
         C ← F/E
         DAO ← G/E
         U ← C**2+1.0
         R ← 2.0*C*DAO-C*(TABLE-INPUT(2,2)+TABLE-INPUT(3,2))-
            TABLE-INPUT(2,1)-TABLE-INPUT(3,1)
         SAL ← DAO**2-DAO*(TABLE-INPUT(2,2)+TABLE-
            INPUT(3,2))+TABLE-INPUT(2,2)*TABLE-
            INPUT(3,2)+TABLE-INPUT(2,1)*TABLE-INPUT(3,1)
         DISC ← SQRT(R**2-4.0*U*SAL)
         X01 ← (-R+DISC)/(2.0*U)
         X02 ← (-R-DISC)/(2.0*U)
         Y01 ← C*X01+DAO
         Y02 ← C*X02+DAO
      END IF
   END IF
   ALGORITHM SELECTION(TABLE-INPUT,X01,X02,Y01,Y02)
      IF TABLE-INPUT(1,3)≠(PI/2.0) THEN
         VALOR 1 ← ((TABLE-INPUT(2,1)-X01)*(TABLE-INPUT(1,2)-
            Y01)-(TABLE-INPUT(1,1)-X01)*(TABLE-INPUT(2,2)-
```

```
            Y01))/((TABLE-INPUT(2,2)-Y01)*(TABLE-INPUT(1,2)-
            Y01)+(TABLE-INPUT(2,1)-X01)*(TABLE-INPUT(1,1)-
            X01))
         VALOR 2 ←((TABLE-INPUT(2,1)-X02)*(TABLE-INPUT(1,2)-
            Y02)-(TABLE-INPUT(1,1)-X02)*(TABLE-INPUT(2,2)-
            Y02))/((TABLE-INPUT(2,2)-Y02)*(TABLE-INPUT(1,2)-
            Y02)+(TABLE-INPUT(2,1)-X02)*(TABLE-INPUT(1,1)
            -X02))
         MODUL 1 ←ABS(TANGENT(TABLE-INPUT(1,3))-VALOR 1)
         MODUL 2 ←ABS(TANGENT(TABLE-INPUT(1,3))-VALOR 2)
         IF MODUL 1 < MODUL 2 THEN
            X0 ← X01
            Y0 ← Y01
         ELSE
            X0 ←X02
            Y0 ←Y02
      END IF
   ELSE IF TABLE-INPUT(2,3)≠(PI/2.0) THEN
      VALOR 1 ←((TABLE-INPUT(3,1)-X01)*(TABLE-INPUT(2,2)-
         Y01)-(TABLE-INPUT(2,1)-X01)*(TABLE-INPUT(3,2)-
         Y01))/((TABLE-INPUT(3,2)-Y01)*(TABLE-INPUT(2,2)-
         Y01)+(TABLE-INPUT(3,1)-X01)*TABLE-INPUT(2,1)-X01))
      VALOR 2 ←((TABLE-INPUT(3,1)-X02)*(TABLE-INPUT(2,2)-
         Y02)-(TABLE-INPUT(2,1)-X02)*(TABLE-INPUT(3,2)-
         Y02))/((TABLE-INPUT(3,2)-Y02)*(TABLE-INPUT(2,2)-
         Y02)+(TABLE-INPUT(3,1)-X02)*(TABLE-INPUT(2,1)-X02))
      MODUL 1←ABS(TANGENT(TABLE-INPUT(2,3))-VALOR 1)
      MODUL 2 ←ABS(TANGENT(TABLE-INPUT(2,3))-VALOR 2)
      IF MODUL 1 < MODUL 2 THEN
         X0← X01
         Y0 ←Y01
      ELSE
         X0 ←X02
         Y0 ←Y02
      END IF
   ELSE
      VALOR 1 ←(TABLE-INPUT(2,2)-Y01)*(TABLE-INPUT(1,2)-
         Y01)+(TABLE-INPUT(2,1)-X01)*(TABLE-INPUT(1,1)-X01)
      VALOR 2 ←(TABLE-INPUT(2,2)-Y02)*(TABLE-INPUT(1,2)-
         Y02)+(TABLE-INPUT(2,1)-X02)*(TABLE-INPUT(1,1)-X02)
      MODUL 1←ABS(VALOR 1)
      MODUL 2 ←ABS(VALOR 2)
      IF MODUL 1 <MODUL 2  THEN
         X0 ←X01
         Y0 ← Y01
      ELSE
         X0 ←X02
         Y0 ←Y02
      END IF
   END IF
```

```
      END SELECTION (XO,YO)
   END INITIAL-COORDINATES(XO,YO)

MODULE 36
ALGORITHM RESTORE-INITIAL-DATA(TABLE-INPUT,STORE)
   TABLE-INPUT(1,1) ← STORE(1)
   TABLE-INPUT(1,2) ← STORE(2)
   TABLE-INPUT(1,3) ← STORE(3)
   TABLE-INPUT(2,1) ← STORE(4)
   TABLE-INPUT(2,2) ← STORE(5)
   TABLE-INPUT(2,3) ← STORE(6)
   TABLE-INPUT(3,1) ← STORE(7)
   TABLE-INPUT(3,2) ← STORE(8)
   TABLE-INPUT(I,1) ← STORE(9)
   TABLE-INPUT(I,2) ← STORE(10)
   TABLE-INPUT(I,3) ← STORE(11)
   TABLE-INPUT(J,1) ← STORE(12)
   TABLE-INPUT(J,2) ← STORE(13)
   TABLE-INPUT(J,3) ← STORE(14)
   TABLE-INPUT(K,1) ← STORE(15)
   TABLE-INPUT(K,2) ← STORE(16)
END RESTORE-INITIAL-DATA(TABLE-INPUT)

MODUDULE 37
ALGORITHM INITIAL-AZIMUTHS(XO,YO,TABLE-INPUT,M)
   DO FOR I ← 1 TO M
      IF YO=TABLE-INPUT(I,2) AND XO > TABLE-INPUT(I,1) THEN
         LIST-AZ(I) ← (3.0*PI)/2.0
      ELSE IF YO=TABLE-INPUT(I,2) AND XO < TABLE-INPUT(I,1) THEN
         LIST-AZ(I) ← PI/2.0
      ELSE
         LIST-ALFA(I) ← ARC TANGENT((TABLE-INPUT(I,1)-XO)/
                                    (TABLE-INPUT(I,2)-YO))
         IF LIST-ALFA(I) ≥ 0.0 AND XO < TABLE-INPUT(I,1) THEN
            LIST-AZ(I) ← LIST-ALFA(I)
      ELSE IF LIST-ALFA(I) < 0.0 AND XO > TABLE-INPUT(I.1) THEN
         LIST-AZ(I) ← LIST-ALFA(I)+2.0*PI
      ELSE
         LIST-AZ(I) ← LIST-ALFA(I)+PI
         END IF
      END IF
   END DO
END INITIAL-AZIMUTHS(LIST-AZ)

MODULE 38
ALGORITHM MATRIX-L(TABLE-INPUT(I,3),LIST-AZ,N)
   ALGORITHM ZERO(LIST-L)
      DO FOR I ← 1 TO 10
         LIST-L(I) ← 0.0
      END DO
```

```
   END ZERO(LIST-L)
   DO FOR J←1 TO N
      J←I+1
      LIST-L(I)←TABLE-INPUT(I,3)+LIST-AZ(I)-LIST-AZ(J)
   END DO
END MATRIX-L(LIST-L)

MODULE 39
ALGORITHM SQUARED-DISTANCES(TABLE-INPUT,XO,YO)
   DO FOR I←1 TO M
      LIST-SO(I)←(TABLE-INPUT(I,1)-XO)**2+(TABLE-INPUT(I,2)-
         YO)**2
   END DO
END SQUARED-DISTANCES(LIST-SO)

MODULE 40
ALGORITHM MATRIX-A(N,TABLE-INPUT,XO,YO,LIST-SO)
   ALGORITHM ZERO(TABLE-A)
      DO FOR I←1 TO 10
         DO FOR J←1 TO 2
            TABLE-A(I,J)← 0.0
         END DO
      END DO
   END ZERO(TABLE-A)
   DO FOR I←1 TO N
      J←I+1
      TABLE-A(I,1)←(YO-TABLE-INPUT(J,2))/LIST-SO(J)-
                   (YO-TABLE-INPUT(I,2))/LIST-SO(I)
   END DO
   DO FOR I←1 TO N
      J←I+1
      TABLE-A(I,2)←(YO-TABLE-INPUT(I,1))/LIST-SO(I)-
                   (XO-TABLE-INPUT(J,1))/LIST-SO(J)
   END DO
END MATRIX-A(TABLE-A)
```

```
MODULE 50
ALGORITHM FIX-BY-TWO-RANGES-AND-ONE-AZIMUTH
   N ←3
   PI ←3.141592653589793
   DO FOR I←1 TO 3
      INPUT TABLE-INPUT(I,1),TABLE-INPUT(I,2),TABLE-INPUT(I,4)
   END DO
   DO FOR I←1 TO 2
      OUTPUT'ST#',I,'EAST=',TABLE-INPUT(I,1),'NORT=',TABLE-
         INPUT(I,2),'ST ERROR=',TABLE-INPUT(I,4),'METERS'
   END DO
   OUTPUT'ST#3EAST=',TABLE-INPUT(3,1),'NORT=',TABLE-
         INPUT(3,2),'ST ERROR=',TABLE-INPUT(3,4),'DEGREES'
   ALGORITHM CONVERSION-DEGREES-RADIANS(TABLE-INPUT(3,4))
      TABLE-INPUT(3,4)←TABLE-INPUT(3,4)*(PI/180.0)
   END CONVERSION-DEGREES-RADIANS(TABLE-INPUT(3,4))
   INPUT TABLE-INPUT(1,3),TABLE-INPUT(2,3),TABLE-INPUT(3,3)
   DO WHILE TABLE-INPUT(1,3)≠0.0
      OUTPUT'OBSERVED RANGE DISTANCES AND AZIMUTH ANGLE'
      OUTPUT'R1=',TABLE-INPUT(1,3),'METERS'
      OUTPUT'R2=',TABLE-INPUT(2,3),'METERS'
      OUTPUT'A=',TABLE-INPUT(3,3),'DEGREES'
      ALGORITHM CONVERSION-DEGREES-RADIANS(TABLE-INPUT(3,3))
         TABLE-INPUT(3,3)←TABLE-INPUT(3,3)*(PI/180.0)
      END CONVERSION-DEGREES-RADIANS(TABLE-INPUT(3,3))
      MODULES 51,52,5,53,7
      INPUT TABLE-INPUT(1,3),TABLE-INPUT(2,3),TABLE-INPUT(3,3)
   END DO
END FIX-BY-TWO-RANGES-AND-ONE-AZIMUTH

MODULE 51
ALGORITHM FIRST-INITIAL-POINT(TABLE-INPUT)
   E←TABLE-INPUT(1,3)**2-TABLE-INPUT(2,3)**2+
      TABLE-INPUT(2,2)**2-TABLE-INPUT(1,2)**2+
      TABLE-INPUT(2,1)**2-TABLE-INPUT(1,1)**2
   IF TABLE-INPUT(2,1)≠TABLE-INPUT(1,1) THEN
      E1 ←((TABLE-INPUT(1,2)-TABLE-INPUT(2,2))/
          (TABLE-INPUT(2,1)-TABLE-INPUT(1,1)))**2+1.0
      E2 ←(E*(TABLE-INPUT(1,2)-TABLE-INPUT(2,2)))/
         ((TABLE-INPUT(2,1)-TABLE-INPUT(1,1))**2)-
         2.0*TABLE-INPUT(1,1)*((TABLE-INPUT(1,2)-TABLE-
         INPUT(2,2))/(TABLE-INPUT(2,1)-TABLE-INPUT(1,1)))-2.0*
         TABLE-INPUT(1,2)
      E3 ← (E/(2.0*(TABLE-INPUT(2,1)-TABLE-INPUT(1,1))))**2-
         (E*TABLE-INPUT(1,1)) /(TABLE-INPUT(2,1)-TABLE-
         INPUT(1,1))-TABLE-INPUT(1,3)**2+TABLE-INPUT(1,1)**2+
         TABLE-INPUT(1,2)**2
```

```
E4 ← E2**2-4.0*E1*E3
IF E4 < 0.0 THEN
   XO ← TABLE-INPUT(1,1)+TABLE-INPUT(1,3)*(TABLE-INPUT(2,1)-
       TABLE-INPUT(1,1))/SQRT((TABLE-INPUT(2,1)-
       TABLE-INPUT(1,1)**2+(TABLE-INPUT(2,2)-
       TABLE-INPUT(1,2))**2)
   YO ←TABLE-INPUT(1,2)+TABLE-INPUT(1,3)*(TABLE-INPUT(2,2)-
       TABLE-INPUT(1,2))/SQRT((TABLE-INPUT(2,1)-
       TABLE-INPUT(1,1))**2+(TABLE-INPUT(2,2)-
       TABLE-INPUT(1,2))**2)
ELSE E4=0.0 THEN
   YO ← -E2/(2.0*E1)
   XO ←E/(2.0*(TABLE-INPUT(2,1)-TABLE-INPUT(1,1)))+
       YO*(TABLE-INPUT(1,2)-TABLE-INPUT(2,2))/
       (TABLE-INPUT(2,1)-TABLE-INPUT(1,1))
ELSE
   YO1 ←(-E2+SQRT(E4))/2.0*E1)
   XO1 ←E/(2.0*(TABLE-INPUT(2,1)-TABLE-INPUT(1,1)))+
        YO1*(TABLE-INPUT(1,2)-TABLE-INPUT(2,2))/
            (TABLE-INPUT(2,1)-TABLE-INPUT(1,1))
   YO2 ←(-E2-SQRT(E4))/(2.0*E1)
   XO2 ←E/(2.0*(TABLE-INPUT(2,1)-TABLE-INPUT(1,1)))+
        YO2*(TABLE-INPUT(1,2)-TABLE-INPUT(2,2))/
            (TABLE-INPUT(2,1)-TABLE-INPUT(1,1))
   IF TABLE-INPUT(3,1)=XO1 AND TABLE-INPUT(3,2)=YO1 THEN
      XO ← XO2
      YO ←YO2
   ELSE IF TABLE-INPUT(3,1)=XO2 AND TABLE-INPUT(3,2)=
      YO2 THEN
      XO ←XO1
      YO ←YO1
   ELSE
      CALL CRITERIUM(TABLE-INPUT(3,1),TABLE-INPUT(3,2),
         XO1,YO1,A301)
      CALL CRITERIUM(TABLE-INPUT(3,1),TABLE-INPUT(3,2),
         XO2,YO2,A302)
      IF A301=TABLE-INPUT(3,3) AND A301=A302 THEN
         OUTPUT'SOLUTION UNDETERMINED FOR THAT DATA SET'
         PICK UP ANOTHER DATA SET
      ELSE IF ((A301-TABLE-INPUT(3,3))**2)=
              ((A302-TABLE-INPUT(3,3))**2) THEN
         XO ←(XO1+XO2)/2.0
         YO ←(YO1+YO2)/2.0
      ELSE IF ((A301-TABLE-INPUT(3,3))**2) >
              ((A302-TABLE-INPUT(3,3))**2) THEN
         XO ← XO2
         YO ← YO2
```

```
            ELSE IF ((A301-TABLE-INPUT(3,3))**2) <
                    ((A302-TABLE-INPUT(3,3))**2) THEN
               XO← XO1
               YO ←YO1
            END IF
         END IF
      END IF
   ELSE
      YO ← E/(2.0*(TABLE-INPUT(2,2)-TABLE-INPUT(1,2)))
      F ←TABLE-INPUT(1,3)**2-(TABLE-INPUT(1,2)-YO)**2
      IF F ≤0.0 THEN
         XO ←TABLE-INPUT(1,1)
      ELSE
         XO1 ←TABLE-INPUT(1,1)+SQRT(F)
         YO1 ←YO
         XO2 ←TABLE-INPUT(1,1)-SQRT(F)
         YO2 ←YO
         IF TABLE-INPUT(3,1)=XO1 AND TABLE-INPUT(3,2)=YO1 THEN
            XO ←XO2
         ELSE IF TABLE-INPUT(3,1)=XO2 AND TABLE-INPUT(3,2)=
            YO2 THEN
            XO←XO1
         ELSE
            CALL CRITERIUM(TABLE-INPUT(3,1),TABLE-INPUT(3,2),
               XO1,YO1,A301)
            CALL CRITERIUM(TABLE-INPUT(3,1),TABLE-INPUT(3,2),
               XO2,YO2,A302)
            IF A301=TABLE-INPUT(3,3) AND A301=A302 THEN
               OUTPUT'SOLUTION IS UNDETERMINED FOR THAT DATA SET'
               PICK UP ANOTHER DATA SET
            ELSE IF ((A301-TABLE-INPUT(3,3))**2)=
                    ((A302-TABLE-INPUT(3,3))**2) THEN
               XO←XO1
            ELSE IF ((A301-TABLE-INPUT(3,3))**2) >
                    ((A302-TABLE-INPUT(3,3))**2)THEN
               XO←XO2
            ELSE IF ((A301-TABLE-INPUT(3,3))**2) <
                    ((A302-TABLE-INPUT(3,3))**2) THEN
               XO← XO1
            END IF
         END IF
      END IF
   END IF
END FIRST INITIAL-POINT(XO,YO)

MODULE 52
ALGORITHM ITERATIONS (TOLERANCE)
   DO UNTIL TOLERANCE <1.0
      MODULES 54,55,56,57,58,2,59,12,13,14
   END DO
END ITERATIONS(XO,YO)
```

```
MODULE 53
ALGORITHM PRECISION(TABLE-A,TABLE-WEIGHT,TABLE-Q,LIST-L,DELTA-X,
                    DELTA Y,N,S30)
    MODULES 18,19,60,21,22
END PRECISION (SU,SX,SY,SXY,RO)

MODULE 54
ALGORITHM A30(TABLE-INPUT,XO,YO)
    CALL CRITERIUM(TABLE-INPUT(3,1),TABLE-INPUT(3,2),XO,YO,A30)
END A30(A30)

MODULE 55
ALGORITHM DISTANCES(TABLE-INPUT,XO,YO)
    S10←SQRT((TABLE-INPUT(1,1)-XO)**2+(TABLE-INPUT(1,2)-
       YO)**2)
    S20 ←SQRT((TABLE-INPUT(2,1)-XO)**2+(TABLE-INPUT(2,2)-
       YO)**2)
    S30←SQRT((TABLE-INPUT(3,1)-XO)**2+(TABLE-INPUT(3,2)-
       YO)**2)
END DISTANCES

MODULE 56
ALGORITHM MATRIX-A(TABLE-INPUT,XO,YO,S10,S20,S30,A30)
    TABLE-A(1,1)←(XO-TABLE-INPUT(1,1))/S10
    TABLE-A(1,2)←(YO-TABLE-INPUT(1,2))/S10
    TABLE-A(2,1)←(XO-TABLE-INPUT(2,1))/S20
    TABLE-A(2,2) ←(YO-TABLE-INPUT(2,2))/S20
    TABLE-A(3,1)←COSINE(A30-TABLE-INPUT(3,3))*(YO-
       TABLE-INPUT(3,2))/S30+SINE(A30-TABLE-INPUT(3,3))*
       (XO-TABLE-INPUT(3,1))/S30
    TABLE-A(3,2)←COSINE(A30-TABLE-INPUT(3,3))*(TABLE-
       INPUT(3,1)-XO)/S30+SINE(A30-TABLE-INPUT(3,3))*(YO-
       TABLE-INPUT(3,2)))/S30
END MATRIX-A(TABLE-A)

MODULE 57
ALGORITHM LIST-L(TABLE-INPUT,S10,S20,S30,A30)
    LIST-L(1)← TABLE-INPUT(1,3)-S10
    LIST-L(2) ←TABLE-INPUT(2,3)-S20
    LIST-L(3)←SINE(TABLE-INPUT(3,3)-A30)*S30
END LIST-L(LIST-L)

MODULE 58
ALGORITHM BEFORE-WEIGHT-MATRIX(TABLE-INPUT(3,4),S30)
    TABLE-INPUT(3,4)← S30*SINE(TABLE-INPUT(3,4))
END BEFORE-WEIGHT-MATRIX(TABLE-INPUT(3,4))
```

```
MODULE 59
ALGORITHM AFTER-WEIGHT-MATRIX(TABLE-INPUT(3,4),S30)
    TABLE-INPUT(3,4)←ARC SINE(TABLE-INPUT(3,4)/S30)
END AFTER-WEIGHT-MATRIX(TABLE-INPUT(3,4))

MODULE 60
ALGORITHM ST-DEVIATION-OF-EACH-OBS(SU,TABLE-WEIGHT,S30)
    OUTPUT'PRECISION OF OBSERVATIONS'
    DO FOR I←1 TO 2
       S←(SU/SQRT(TABLE-WEIGHT(I,J)))
       OUTPUT'ST DEVIATION OF OBS',I,'=',S,'METERS'
    END DO
    S←ARC SINE(SU/(SQRT(TABLE-WEIGHT(3,3))*S30))*(180.0/PI)
    OUTPUT'ST DEVIATION OF OBS 3=',S,'DEGREES'
END ST-DEVIATION-OF-EACH-OBS

MODULE 70
SUBROUTINE CRITERIUM(XS,YS,XP,YP,ASP)
    PI←3.14159 26535 89793
    IF YP=YS AND XP > XS THEN
       ASP←PI/2.0
    ELSE IF YP=YS AND XP < XS THEN
       ASP←3.0*PI/2.0
    ELSE
       ALFA←ARC TANGENT((XP-XS)/YP-YS))
       IF ALFA ≥ 0.0 AND XP ≥ XS THEN
          ASP←ALFA
       ELSE IF ALFA < 0.0 AND XP < XS THEN
          ASP←ALFA+2.0*PI
       ELSE
          ASP←ALFA+PI
       END IF
    END IF
    RETURN
END CRITERIUM
```

```
NUMBER OF STATIONS= 3
ST# 1    EAST=    595794.50   NORT=   4055042.70   ST ERROR= 0.0200
ST# 2    EAST=    597967.80   NORT=   4053453.20   ST ERROR= 0.0240
ST# 3    EAST=    603425.20   NORT=   4053917.20   ST ERROR= 0.0180


OBSERVED AZIMUTHS
 AZIMUTH FROM STATION# 1 = 76.017   DEGREES
 AZIMUTH FROM STATION# 2 = 45.541   DEGREES
 AZIMUTH FROM STATION# 3 =313.005   DEGREES


ADJUSTED COORDINATES X=    600868.306    Y=   4056302.781


PRECISION OF OBSERVATIONS
 ST DEVIATION OF OBS 1 =0.031  DEGREES
 ST DEVIATION OF OBS 2 =0.038  DEGREES
 ST DEVIATION OF OBS 3 =0.028  DEGREES
 SX=  2.13    SY=  1.70    SXY=   -0.862
 CORRELATION COEFFICIENT RO=-.24


ERROR ELIPSE SEMI-AXIS AND ORIENTATION
SEMI-MAJOR AXIS SA= 2.283
SEMI-MINOR AXIS SB= 1.636
ANGLE FROM X-AXIS TO SA ANTICLOCKWISE=157.0DEG
```

```
NUMBER CF SEXTANT ANGLES= 3
ST# 1    EAST=    603425.20    NORT=   4053917.20    ST ERROR= 1.0000
ST# 2    EAST=    600372.00    NORT=   4051216.90    ST ERROR= 1.0000
ST# 3    EAST=    597967.80    NORT=   4053453.20    ST ERROR= 1.0000
ST# 4    EAST=    595794.50    NORT=   4055042.70


OBSERVED SEXTANT ANGLES
 SEXTANT ANGLE BETWEEN ST# 1 AND ST# 2 = 49.927 DEGREES
 SEXTANT ANGLE BETWEEN ST# 2 AND ST# 3 = 38.130 DEGREES
 SEXTANT ANGLE BETWEEN ST# 3 AND ST# 4 = 30.396 DEGREES


ADJUSTED CCORDINATES X=    600864.586    Y=   4056512.323


PRECISION CF OBSERVATIONS
 ST DEVIATION CF OBS 1 =0.007   DEGREES
 ST DEVIATION CF OBS 2 =0.007   DEGREES
 ST DEVIATION CF OBS 3 =0.007   DEGREES
 SX=  1.02    SY=  0.48    SXY=   -0.097
 CORRELATICN CCEFFICIENT RO=-.20


ERROR ELIPSE SEMI-AXIS AND ORIENTATION
SEMI-MAJCR AXIS SA=  1.050
SEMI-MINOR AXIS SB=  0.480
ANGLE FRCM X-AXIS TO SA ANTICLOCKWISE=173.3DEG
```

```
ST# 1    EAST=    595794.50    NORT=   4055042.70    ST ERROR= 10.000   METERS
ST# 2    EAST=    603425.20    NORT=   4053917.20    ST ERROR= 10.000   METERS
ST# 3    EAST=    597967.80    NORT=   4053453.20    ST ERROR=  0.024   DEGREES


OBSERVED RANGE DISTANCES AND AZIMUTH ANGLE
 R1=  5233.00  METERS
 R2=  3515.CO  METERS
 A=     45.54  DEGREES


ADJUSTED CCORDINATES X=   600872.166   Y=  4056304.120


PRECISION CF CBSERVATIONS
 ST DEVIATICN CF OBS 1  =  3.45  METERS
 ST DEVIATICN CF OBS 2  =  3.45  METERS
 ST DEVIATICN OF CBS 3  = 0.008  DEGREES
 SX=  2.86   SY=  2.88   SXY=    7.911
 CORRELATICN CCEFFICIENT RC= 0.96


 ERROR ELIFSE SEMI-AXIS AND ORIENTATION
 SEMI-MAJOR AXIS SA=14.265
 SEMI-MINOR AXIS SB= 2.051
 ANGLE FROM X-AXIS TO SA ANTICLOCKWISE= 45.2DEG
```

```
// EXEC FORTXCLG
//FORT.SYSIN DD *
C PROGRAM FOR FIX DETERMINATION GIVEN N AZIMUTHS FROM DIFFERENT STATIONS
C
C THIS PROGRAM DETERMINES ADJUSTED COORDINATES OF VESSEL,USING LEAST
C SQUARES METHOD.ALSO GIVES INFORMATION ABOUT PRECISION OF OBSERVED
C AZIMUTHS AND COMPUTED RESULTS,INCLUDING ERROR ELIPSE.
C
C USER'S INSTRUCTIONS
C 1-INPUT NUMBER N OF STATIONS USING FORMAT 100 (MAXIMUN N=10)
C 2-FOR EACH STATION INPUT RESPECTIVE X-COORDINATE(EASTING),Y-COORDINATE
C (NORTHING),AND STANDARD DEVIATION OF OBSERVED AZIMUTHS USING
C FORMAT 178.IF THERE IS NO INFORMATION ABOUT STANDARD ERRORS,ENTER 1.0
C 3-PRESERVING THE ORDER ESTABLISHED FOR THE N STATIONS,INPUT THE N
C OBSERVED AZIMUTHS(ONE IN EACH CARD)USING FORMAT 180
C 4-WHEN NO MORE DATA SETS ARE AVAILABLE INPUT A 'DUMMY' DATA SET WITH
C ALL VALUES EQUAL TO 400.0 USING THE SAME FORMAT 180
C
C ALGORITHM FIX BY N AZIMUTHS
      INTEGER N,I,J,K
      REAL*8 PI,GREAT,TBINP(10,4),TBW(10,10),M1,MK,XO,YO,DTAN,DATAN,
     1       AO(10),ALFA(10),L(10),SO(10),A(10,2),ATW(2,10),ATWA(2,2)
     2      ,Q(2,2),BETA,ATWL(2),DELTAX,DELTAY,TOL,X(2),AX(10),V(10),
     3       VTW(10),VTWV,TRACE,SU,CHARLE,DSQRT,S,SX,SY,SXY,RO,D,SA,
     4       SB,GAMA,OMEGA,X1,Y1,D1,GAMAO,X10,X11,AVER
  100 FORMAT(1X,I2)
  101 FORMAT('1',1X,'NUMBER OF STATIONS=',I2)
  106 FORMAT('0',1X,'POSITION IS UNDETERMINED FOR THAT DATA SET')
  126 FORMAT(//////,1X,'ADJUSTED COORDINATES X=',F13.3,3X,'Y=',F13.3)
  134 FORMAT(//////,1X,'PRECISION OF OBSERVATIONS')
  135 FORMAT('0',1X,'ST DEVIATION OF OBS',I2,' =',F5.3,2X,'DEGREES')
  136 FORMAT('0',1X,'SX=',F6.2,3X,'SY=',F6.2,3X,'SXY=',F9.3)
  137 FORMAT('0',1X,'CORRELATION COEFFICIENT RO=',F4.2)
  138 FORMAT(//////,1X,'ERROR ELIPSE SEMI-AXIS AND ORIENTATION')
  139 FORMAT(/,1X,'SEMI-MAJOR AXIS SA=',F6.3)
  140 FORMAT(/,1X,'SEMI-MINOR AXIS SB=',F6.3)
  142 FORMAT(/,1X,'ANGLE FROM X-AXIS TO SA ANTICLOCKWISE=',F5.1,'DEG')
  178 FORMAT(1X,2F12.2,F7.4)
  179 FORMAT('0',1X,'ST#',I2,3X,'EAST=',F12.2,3X,'NORT=',F12.2,3X,
     1       'ST ERROR=',F7.4)
  180 FORMAT(1X,F9.5)
  181 FORMAT(//////,1X,'OBSERVED AZIMUTHS')
  182 FORMAT('0',1X,'AZIMUTH FROM STATION#',I2,' =',F7.3,'   DEGREES')
  183 FORMAT('1',1X)
C  (N=NUMBER OF STATIONS(MAXIMUN=10).FOR EACH STATION INPUT X-COORDINATE
C   (EASTING),Y-COORDINATE(NORTHING) AND STANDARD ERROR )
      READ(5,100)N
      PI=3.141592653589793
```

```
      WRITE(6,101)N
      DO 10 I=1,N
       READ(5,178)TBINP(I,1),TBINP(I,2),TBINP(I,4)
      WRITE(6,179)I,TBINP(I,1),TBINP(I,2),TBINP(I,4)
   10 CONTINUE
      DO 249 I=1,N
       READ(5,180)TBINP(I,3)
  249 CONTINUE
  250 IF(TBINP(1,3).EQ.400.0)GO TO 999
       WRITE(6,181)
       DO 252 I=1,N
        WRITE(6,182)I,TBINP(I,3)
C       ALGORITHM CONVERSION_DEGREES_RADIANS(TBINP(I,3))
         TBINP(I,3)=TBINP(I,3)*(PI/180.0)
C       END CONVERSION_DEGREES_RADIANS(TBINP(I,3))
  252  CONTINUE
C ALGORITHM WEIGHT_MATRIX(N,TBINP(I,4))
C     ALGORITHM ZERO(TBW)
       DO 11 I=1,10
        DO 12 J=1,10
         TBW(I,J)=.000000000
   12   CONTINUE
   11  CONTINUE
C     END ZERO(TBW)
C     ALGORITHM SQUARE(N,TBINP(I,4),TBW)
       DO 13 I=1,N
        TBW(I,I)=TBINP(I,4)**2
   13  CONTINUE
C     END SQUARE(TBW)
C     ALGORITHM NORMALIZE(TBW)
       GREAT=TBW(1,1)
       DO 14 I=2,N
        IF(TBW(I,I).GT.GREAT)GREAT=TBW(I,I)
   14  CONTINUE
       DO 15 I=1,N
        TBW(I,I)=GREAT/TBW(I,I)
   15  CONTINUE
C     END NORMALIZE(TBW)
C END WEIGHT_MATRIX(TBW)
C ALGORITHM FIRST_INITIAL_POINT(TBINP,N)
C     ALGORITHM SELECT AZIMUTHS(TBINP(I,3),N)
      I=2
   19  IF(DTAN(TBINP(I,3)).NE.DTAN(TBINP(1,3)))GO TO 17
        I=I+1
        IF(I.LE.N)GO TO 18
        WRITE(6,106)
      GO TO 998
   18   CONTINUE
```

```
      GO TO 19
   17 CONTINUE
C     END SELECT_AZIMUTHS(I)
C     ALGORITHM INITIAL_COORDINATES(TBINP,I)
      IF((TBINP(1,3).NE.0.0).AND.(TBINP(1,3).NE.PI))GO TO 20
       MK=DTAN((5./2.)*PI-TBINP(I,3))
       XO=TBINP(1,1)
       YO=TBINP(I,2)+MK*(XO-TBINP(I,1))
      GO TO 22
   20 IF((TBINP(I,3).NE.0.0).AND.(TBINP(I,3).NE.PI))GO TO 21
       M1=DTAN((5./2.)*PI-TBINP(1,3))
       XO=TBINP(I,1)
       YO=TBINP(1,2)+M1*(XO-TBINP(1,1))
       GO TO 22
   21 CONTINUE
       M1=DTAN((5./2.)*PI-TBINP(1,3))
       MK=DTAN((5./2.)*PI-TBINP(I,3))
       XO=(TBINP(I,2)-TBINP(1,2)+M1*TBINP(1,1)-MK*TBINP(I,1))/(M1-MK)
       YO=TBINP(1,2)+M1*(XO-TBINP(1,1))
   22 CONTINUE
C     END INITIAL_COORDINATES(XO,YO)
C END FIRST_INITIAL_POINT(XO,YO)
C ALGORITHM ITERATIONS(TOL)
   51 CONTINUE
C ALGORITHM INITIAL_AZIMUTHS(XO,YO,TBINP,N)
      DO 23 I=1,N
      IF((YO.NE.TBINP(I,2)).OR.(XO.LE.TBINP(I,1)))GO TO 24
       AO(I)=PI/2.000000000
      GO TO 30
   24 IF((YO.NE.TBINP(I,2)).OR.(XO.GE.TBINP(I,1)))GO TO 25
       AO(I)=(3.000000000*PI)/2.000000000
      GO TO 30
   25 IF((XO.NE.TBINP(I,1)).OR.(YO.LE.TBINP(I,2)))GO TO 26
       AO(I)=0.000000000
      GO TO 30
   26 IF((XO.NE.TBINP(I,1)).OR.(YO.GE.TBINP(I,2)))GO TO 27
       AO(I)=PI
      GO TO 30
   27 CONTINUE
       ALFA(I)=DATAN((XO-TBINP(I,1))/(YO-TBINP(I,2)))
       IF((ALFA(I).LE.0.000).OR.(XO.LE.TBINP(I,1)))GO TO 28
        AO(I)=ALFA(I)
       GO TO 29
   28  IF((ALFA(I).LE.0.00).OR.(XO.GE.TBINP(I,1)))GO TO 31
        AO(I)=ALFA(I)+PI
       GO TO 29
   31  IF((ALFA(I).GE.0.00).OR.(XO.LE.TBINP(I,1)))GO TO 32
         AO(I)=ALFA(I)+PI
```

```
      GO TO 29
   32 CONTINUE
       AO(I)=ALFA(I)+2.000000000*PI
   29 CONTINUE
   30 CONTINUE
   23 CONTINUE
C END INITIAL_AZIMUTHS(AO)
C ALGORITHM MATRIX_L(TBINP(I,3),AO,N)
C     ALGORITHM ZERO(L)
      DO 34 I=1,10
       L(I)=.0000000000
   34 CONTINUE
C     END ZERO(L)
      DO 35 I=1,N
       L(I)=TBINP(I,3)-AO(I)
   35 CONTINUE
C END MATRIX_L(L)
C ALGORITHM INITIAL SQUARED DISTANCES(XO,YO,N,TBINP)
      DO 36 I=1,N
       SO(I)=(XO-TBINP(I,1))**2+(YO-TBINP(I,2))**2
   36 CONTINUE
C END INITIAL_SQUARED_DISTANCES(SO)
C ALGORITHM MATRIX_A(N,TBINP,XO,YO,SO)
C     ALGORITHM ZERO(A)
      DO 38 I=1,10
       DO 39 J=1,2
        A(I,J)=.000000000
   39  CONTINUE
   38 CONTINUE
C     END ZERO(A)
C     ALGORITHM ELEMENTS(A,TBINP,XO,YO,SO)
       DO 40 I=1,N
        A(I,1)=(YO-TBINP(I,2))/SO(I)
   40  CONTINUE
       DO 41 I=1,N
        A(I,2)=(TBINP(I,1)-XO)/SO(I)
   41  CONTINUE
C      END ELEMENTS(A)
C END MATRIX_A(A)
C ALGORITHM NORMAL_EQUATIONS(A,W,L)
C ALGORITHM TRANSPOSE(A)*TBW(A,TBW)
      DO 43 I=1,2
       DO 44 J=1,10
        ATW(I,J)=.0000000000
        DO 45 K=1,10
         ATW(I,J)=ATW(I,J)+A(K,I)*TBW(K,J)
   45   CONTINUE
   44  CONTINUE
```

```
   43 CONTINUE
C END TRANSPOSE(A)*TBW(ATW)
C ALGORITHM MATRIX_ATWA(ATW,A)
      DO 46 I=1,2
       DO 47 J=1,2
        ATWA(I,J)=.0000000000
        DO 48 K=1,10
         ATWA(I,J)=ATWA(I,J)+ATW(I,K)*A(K,J)
   48   CONTINUE
   47  CONTINUE
   46 CONTINUE
C END MATRIX_ATWA(ATWA)
C ALGORITHM INVERT_ATWA(ATWA)
      BETA=ATWA(1,2)**2-ATWA(1,1)*ATWA(2,2)
      Q(1,1)=-ATWA(2,2)/BETA
      Q(1,2)=ATWA(1,2)/BETA
      Q(2,1)=Q(1,2)
      Q(2,2)=-ATWA(1,1)/BETA
C END INVERT_ATWA(Q)
C ALGORITHM MATRIX_ATWL(ATWL)
      DO 49 I=1,2
       ATWL(I)=.0000000000
       DO 50 K=1,10
        ATWL(I)=ATWL(I)+ATW(I,K)*L(K)
   50  CONTINUE
   49 CONTINUE
C END MATRIX_ATWL(ATWL)
C ALGORITHM ADJUSTED_INCREMENTS(Q,ATWL)
      DELTAX=Q(1,1)*ATWL(1)+Q(1,2)*ATWL(2)
      DELTAY=Q(2,1)*ATWL(1)+Q(2,2)*ATWL(2)
C END ADJUSTED_INCREMENTS(DELTAX,DELTAY)
C END NORMAL_EQUATIONS(DELTAX,DELTAY)
C ALGORITHM NEW_INITIAL_POINT(XO,YO,DELTAX,DELTAY)
      XO=XO+DELTAX
      YO=YO+DELTAY
C END NEW_INITIAL_POINT(XO,YO)
C ALGORITHM TOLERANCE(DELTAX,DELTAY)
      TOL=DELTAX**2+DELTAY**2
C END TOLERANCE(TOL)
      IF(TOL.GE.1.000)GO TO 51
C END ITERATIONS(XO,YO)
C ALGORITHM FINAL_ADJUSTED_POSITION(XO,YO)
      WRITE(6,126)XO,YO
C FINAL_ADJUSTED_POSITION(X,Y)
C ALGORITHM PRECISION(A,TBW,Q,L,DELTAX,DELTAY,N)
C ALGORITHM RESIDUALS(A,L,DELTAX,DELTAY,N)
      X(1)=DELTAX
      X(2)=DELTAY
```

```
C       ALGORITHM AX(A,X,N)
        DO 52 I=1,N
          AX(I)=.000000000
          DO 53  J=1,2
            AX(I)=AX(I)+A(I,J)*X(J)
   53     CONTINUE
   52   CONTINUE
C       END AX(AX)
C   ALGORITHM V(AX,L,N)
        DO 54 I=1,N
          V(I)=AX(I)-L(I)
   54   CONTINUE
C       END V(V)
C       END RESIDUALS (V)
C   ALGORITHM ST_DEVIATION_OF_UNIT_WEIGHT_OBS(V,TBW,N)
C       ALGORITHM VTW(V,W,N)
          DO 55  I=1,N
            VTW(I)=V(I)*TBW(I,I)
   55     CONTINUE
C       END VTW(VTW)
C       ALGORITHM VTWV(VTW,V)
          VTWV=.000000000
          DO 56  I=1,N
            VTWV=VTWV+VTW(I)*V(I)
   56     CONTINUE
C       END VTWV(VTWV)
C       ALGORITHM TRACE(TBW)
          TRACE=.0000000000
          DO 57  I=1,N
            TRACE=TRACE+TBW(I,I)
   57     CONTINUE
C         END TRACE(TRACE)
C         ALGORITHM SO(VTWV,TRACE)
          CHARLE=VTWV/(TRACE-2.000000000)
          SU=DSQRT(CHARLE)
C       END SO(SO)
C   END ST_DEVIATION_OF_UNIT_WEIGHT_OBS(SO)
C   ALGORITHM ST_DEVIATION_OF_EACH_OBS(SO,TBW)
        WRITE(6,134)
        DO 58 I=1,N
          S=(SU/DSQRT(TBW(I,I)))*(180.00000000/PI)
          WRITE(6,135)I,S
   58   CONTINUE
C   END ST_DEVIATION_OF_EACH_OBS
C   ALGORITHM ST_DEVIATIONS_AND_COVARIANCE_OF_X_AND_Y(SO,Q)
        SX=SU*DSQRT(Q(1,1))
        SY=SU*DSQRT(Q(2,2))
        SXY=(SU**2)*Q(1,2)
```

```
      WRITE(6,136)SX,SY,SXY
C END ST_DEVIATIONS_AND_COVARIANCE_OF_X_AND_Y(SX,SY,SXY)
C ALGORITHM CORRELATION_CEFFICIENT(SX,SY,SXY)
      RO=SXY/(SX*SY)
      WRITE(6,137)RO
C END CORRELATION_COEFFICIENT(RO)
C END PRECISION(SO,SX,SY,SXY,RO)
C ALGORITHM ERROR_ELIPSE(Q,SU)
      WRITE(6,138)
C ALGORITHM D(Q)
      D=DSQRT((Q(1,1)-Q(2,2))**2+4.000000000*(Q(1,2)**2))
C END D(D)
C ALGORITHM SEMI-MAJOR AXIS(SU,Q)
      SA=SU*DSQRT(2.00000000*Q(1,1)*Q(2,2)/(Q(1,1)+Q(2,2)-D))
      WRITE(6,139)SA
C END SEMI-MAJOR AXIS(SA)
C ALGORITHM SEMI-MINOR AXIS(SU,Q)
      SB=SU*DSQRT(2.00000000*Q(1,1)*Q(2,2)/(Q(1,1)+Q(2,2)+D))
      WRITE(6,140)SB
C END SEMI-MINOR AXIS(SB)
C ALGORITHM GAMA(Q)
      IF(Q(1,1).NE.Q(2,2))GO TO 59
        GAMA=PI/4.00000000000
      GO TO 60
   59  CONTINUE
        OMEGA=DATAN(2.00000000*Q(1,2)/(Q(1,1)-Q(2,2)))
        IF(OMEGA.LT.0.000)GO TO 61
         GAMA=OMEGA/2.000000000
        GO TO 62
   61   CONTINUE
          GAMA=(OMEGA+PI)/2.00000000
   62   CONTINUE
   60 CONTINUE
C END GAMA(GAMA)
C ALGORITHM INTERSECTION(SU,Q,GAMA)
      X10=(SU**2)*Q(1,1)*Q(2,2)
      X11=Q(2,2)-2.00000000*Q(1,2)*DTAN(GAMA)+(DTAN(GAMA)**2)*Q(1,1)
      X1=X10/X11
      Y1=(DTAN(GAMA)**2)*X1
C END INTERSECTION(X1,Y1)
C ALGORITHM AVERAGE(SA,SB)
      AVER=((SA+SB)/2.00000000)**2
C END AVERAGE(AVER)
C ALGORITHM SELECTION(AVER,X1,Y1)
      D1=X1+Y1
      IF(D1.LT.AVER)GO TO 63
        GAMAD=GAMA
      GO TO 64
```

```
   63 CONTINUE
        GAMAO=GAMA+PI/2.0000000000
   64 CONTINUE
      GAMAO=GAMAO*(180.00000000/PI)
      WRITE(6,142)GAMAO
C END SELECTION(GAMAO)
C END ERROR_ELIPSE(SA,SB,GAMAO)
  998 CONTINUE
      WRITE(6,183)
      DO 251 I=1,N
       READ(5,180)TBINP(I,3)
  251 CONTINUE
C     (INTRODUCE NEW DATA SET.IF NO MORE DATA IS AVAILABLE THE LAST
C     SET IS WITH ALL AZIMUTHS EQUAL TO 400.0)
      GO TO 250
  999 CONTINUE
C END FIX_BY_N_AZIMUTHS
      STOP
      END
```

```
// EXEC FORTXCLG
//FORT.SYSIN DD *
C PROGRAM FOR FIX DETERMINATION GIVEN N SEXTANT ANGLES BETWEEN
C DIFFERENT STATIONS
C
C THIS PROGRAM DETERMINES ADJUSTED COORDINATES OF VESSEL USING LEAST
C SQUARES METHOD.ALSO GIVES INFORMATION ABOT PRECISION OF OBSERVED
C SEXTANT ANGLES AND COMPUTED RESULTS,INCLUDING ERROR ELIPSE.
C
C USER'S INSTRUCTIONS
C 1-INPUT NUMBER N OF SEXTANT ANGLES USING FORMAT 100(MAXIMUN N=10)
C 2-ORDER STATIONS IN A CLOCKWISE SENSE AROUND VESSEL'S POSITION.
C 3-FOR EACH STATION INPUT X-COORDINATE(EASTING),Y-COORDINATE(NORTHING),
C    AND STANDARD ERROR OF SEXTANT ANGLE OBSERVED BETWEEN THAT STATION
C    AND THE NEXT ONE,USING FORMAT 184.NOTE THAT FOR THE LAST STATION
C    ONLY X AND Y ARE INPUTTED.
C    IF THERE IS NO INFORMATION ABOUT STANDARD ERROR,ENTER 1.0
C 4-PRESERVING THE ORDER ESTABLISHED FOR THE M=N+1 STATIONS,INPUT THE N
C    OBSERVED SEXTANT ANGLES(ONE IN EACH CARD),USING FORMAT 188
C 5-WHEN NO MORE DATA SETS ARE AVAILABLE INPUT A 'DUMMY' DATA SET WITH
C    ALL N VALUES EQUAL TO 400.0 USING SAME FORMAT 188
C
C
      INTEGER N,M,I,J,K
      REAL*8 PI,TBINP(11,4),TBW(10,10),GREAT,AB,BA,E,F,G,DAO,YO1,YO2,
     1         U,R,SAL,DTAN,DISC,XO1,XO2,H,C,XO,YO,VALOR1,VALOR2,MODUL1,
     2         MODUL2,DABS,DSQRT,ALFA(11),AZ(11),DATAN,L(10),SO(11),
     3         A(10,2),ATW(2,10),ATWA(2,2),Q(2,2),BETA,ATWL(2),DELTAX,
     4         DELTAY,TOL,X(2),AX(10),V(10),VTW(10),VTWV,TRACE,SU,
     5         CHARLE,S,SX,SY,SXY,RO,D,SA,SB,GAMA,OMEGA,X1,Y1,D1,GAMAO,
     6         X10,X11,AVER,ANGUL,FRAC1,FRAC2,DCOS,STORE(16)
  100 FORMAT( 1X,I2)
  126 FORMAT(//////,1X,'ADJUSTED COORDINATES X=',F13.3,3X,'Y=',F13.3)
  134 FORMAT(//////,1X,'PRECISION OF OBSERVATIONS')
  135 FORMAT('0',1X,'ST DEVIATION OF OBS',I2,' =',F5.3,2X,'DEGREES')
  136 FORMAT('0',1X,'SX=',F6.2,3X,'SY=',F6.2,3X,'SXY=',F9.3)
  137 FORMAT('0',1X,'CORRELATION COEFFICIENT RO=',F4.2)
  138 FORMAT(//////,1X,'ERROR ELIPSE SEMI-AXIS AND ORIENTATION')
  139 FORMAT(/,1X,'SEMI-MAJOR AXIS SA=',F7.3)
  140 FORMAT(/,1X,'SEMI-MINOR AXIS SB=',F7.3)
  142 FORMAT(/,1X,'ANGLE FROM X-AXIS TO SA ANTICLOCKWISE=',F5.1,'DEG')
  143 FORMAT('1',1X,'NUMBER OF SEXTANT ANGLES=',I2)
  184 FORMAT( 1X,2F12.2,F7.4)
  185 FORMAT('0',1X,'ST#',I2,3X,'EAST=',F12.2,3X,'NORT=',F12.2,3X,
     1         'ST ERROR=',F7.4)
  186 FORMAT( 1X,2F12.2)
  187 FORMAT('0',1X,'ST#',I2,3X,'EAST=',F12.2,3X,'NORT=',F12.2)
  188 FORMAT( 1X,F9.5)
```

```
  189 FORMAT(//////,1X,'OBSERVED SEXTANT ANGLES')
  190 FORMAT('0',1X,'SEXTANT ANGLE BETWEEN ST#',I2,' AND ST#',I2,' =',
     2       F7.3,' DEGREES')
  191 FORMAT('1',1X)
  192 FORMAT(//,1X,'SOLUTION UNDETERMINED FOR THAT DATA SET')
C ALGORITHM FIX_BY_N_SEXTANT_ANGLES
C  (N=NUMBER OF SEXTANT ANGLES.M=N+1=NUMBER OF STATIONS(M MAXIMUN=11).
C   ORDER STATIONS IN A CLOCKWISE SENSE AROUND VESSEL'S POSITION.
C   FOR EACH STATION INPUT X-COORDINATE(EASTING),Y-COORDINATE(NORTHING),
C   AND STANDARD ERROR OF SEXTANT ANGLE BETWEEN THAT STATION AND THE
C   NEXT ONE.)
      PI=3.141592653589793
      READ(5,100)N
      WRITE(6,143)N
      M=N+1
      DO 253  I=1,N
       READ(5,184)TBINP(I,1),TBINP(I,2),TBINP(I,4)
       WRITE(6,185)I,TBINP(I,1),TBINP(I,2),TBINP(I,4)
  253 CONTINUE
      READ(5,186)TBINP(M,1),TBINP(M,2)
      WRITE(6,187)M,TBINP(M,1),TBINP(M,2)
C     (PRESERVING THE ORDER ESTABLISHED FOR THE STATIONS,INPUT THE
C       SEXTANT ANGLES)
      DO 254  I=1,N
       READ(5,188)TBINP(I,3)
  254 CONTINUE
  255 IF(TBINP(1,3).EQ.400.0)GO TO 999
      WRITE(6,189)
      DO 256  I=1,N
       J=I+1
       WRITE(6,190)I,J,TBINP(I,3)
C      ALGORITHM CONVERSION_DEGREES_RADIANS(TBINP(I,3))
        TBINP(I,3)=TBINP(I,3)*(PI/180.0)
C      END CONVERSION_DEGREES_RADIANS(TBINP(I,3))
  256 CONTINUE
C ALGORITHM WEIGHT_MATRIX(N,TBINP(I,4))
C     ALGORITHM ZERO(TBW)
       DO 11 I=1,10
        DO 12 J=1,10
         TBW(I,J)=.000000000
   12   CONTINUE
   11  CONTINUE
C     END ZERO(TBW)
C     ALGORITHM SQUARE(N,TBINP(I,4),TBW)
       DO 13  I=1,N
        TBW(I,I)=TBINP(I,4)**2
   13  CONTINUE
C     END SQUARE(TBW)
```

```
C       ALGORITHM NORMALIZE(TBW)
         GREAT=TBW(1,1)
         DO 14  I=2,N
          IF(TBW(I,I).GT.GREAT)GREAT=TBW(I,I)
   14    CONTINUE
         DO 15  I=1,N
          TBW(I,I)=GREAT/TBW(I,I)
   15    CONTINUE
C       END NORMALIZE(TBW)
C  END WEIGHT_MATRIX(TBW)
C  ALGORITHM FIRST_INITIAL_POINT_FOR_FIX_BY_N_SEXTANT_ANGLES(TBINP)
C  ALGORITHM SELECT_SEXTANT_ANGLES(TBINP)
        J=1
  258   CONTINUE
         J=J+1
         IF(J.LT.M)GO TO 259
          WRITE(6,192)
          GO TO 998
  259   CONTINUE
         I=J-1
         K=J+1
         ANGUL=TBINP(I,3)+TBINP(J,3)
         FRAC1=DCOS(ANGUL)*DSQRT(((TBINP(I,1)-TBINP(J,1))**2+
     1         (TBINP(I,2)-TBINP(J,2))**2)*((TBINP(K,1)-TBINP(J,1))**2+
     2         (TBINP(K,2)-TBINP(J,2))**2))
         FRAC2=(TBINP(I,1)-TBINP(J,1))*(TBINP(J,1)-TBINP(K,1))+
     3         (TBINP(I,2)-TBINP(J,2))*(TBINP(J,2)-TBINP(K,2))
         IF(FRAC1.EQ.FRAC2)GO TO 258
C  END SELECT_SEXTANT_ANGLES(I,J,K)
C  ALGORITHM INTERCHANGE_DATA(TBINP,I,J,K)
        STORE(1)=TBINP(1,1)
        STORE(2)=TBINP(1,2)
        STORE(3)=TBINP(1,3)
        STORE(4)=TBINP(2,1)
        STORE(5)=TBINP(2,2)
        STORE(6)=TBINP(2,3)
        STORE(7)=TBINP(3,1)
        STORE(8)=TBINP(3,2)
        STORE(9)=TBINP(I,1)
        STORE(10)=TBINP(I,2)
        STORE(11)=TBINP(I,3)
        STORE(12)=TBINP(J,1)
        STORE(13)=TBINP(J,2)
        STORE(14)=TBINP(J,3)
        STORE(15)=TBINP(K,1)
        STORE(16)=TBINP(K,2)
        TBINP(1,1)=STORE(9)
        TBINP(1,2)=STORE(10)
```

```
      TBINP(1,3)=STORE(11)
      TBINP(2,1)=STORE(12)
      TBINP(2,2)=STORE(13)
      TBINP(2,3)=STORE(14)
      TBINP(3,1)=STORE(15)
      TBINP(3,2)=STORE(16)
C END INTERCHANGE_DATA(TBINP,STORE)
C ALGORITHM INITIAL_COORDINATES(TBINP)
      IF(TBINP(1,3).EQ.(PI/2.0).OR.TBINP(2,3).EQ.(PI/2.0))GO TO 67
       AB=DTAN(TBINP(1,3))
       BA=DTAN(TBINP(2,3))
       E=(TBINP(2,1)-TBINP(1,1))/AB+(TBINP(2,1)-TBINP(3,1))/BA+
     2                             TBINP(3,2)-TBINP(1,2)
       F=(TBINP(2,2)-TBINP(1,2))/AB+(TBINP(2,2)-TBINP(3,2))/BA+
     3                             TBINP(1,1)-TBINP(3,1)
       G=(TBINP(1,2)*TBINP(2,1)-TBINP(2,2)*TBINP(1,1))/AB+
     4   (TBINP(2,1)*TBINP(3,2)-TBINP(3,1)*TBINP(2,2))/BA+
     5    TBINP(2,1)*TBINP(3,1)+TBINP(2,2)*TBINP(3,2)-
     6    TBINP(1,1)*TBINP(2,1)-TBINP(1,2)*TBINP(2,2)
       IF(F.NE.0.0)GO TO 68
        DAO=G/E
        YO1=DAO
        YO2=DAO
        U=AB
        R=TBINP(1,2)-TBINP(2,2)-AB*(TBINP(1,1)+TBINP(2,1))
        SAL=AB*(DAO**2-DAO*(TBINP(1,2)+TBINP(2,2))+TBINP(1,1)*TBINP(2,1)
     7         +TBINP(1,2)*TBINP(2,2))+DAO*(TBINP(2,1)-TBINP(1,1))+
     8          TBINP(1,1)*TBINP(2,2)-TBINP(2,1)*TBINP(1,2)
        DISC=DSQRT(R**2-4.0*U*SAL)
        XO1=(-R+DISC)/(2.0*U)
        XO2=(-R-DISC)/(2.0*U)
       GO TO 69
   68  IF(E.NE.0.0)GO TO 70
        H=(-G/F)
        XO1=H
        XO2=H
        U=AB
        R=TBINP(2,1)-TBINP(1,1)-AB*(TBINP(1,2)+TBINP(2,2))
        SAL=AB*(H**2-H*(TBINP(1,1)+TBINP(2,1))+TBINP(1,1)*TBINP(2,1)+
     9          TBINP(1,2)*TBINP(2,2))+H*(TBINP(1,2)-TBINP(2,2))+
     1          TBINP(1,1)*TBINP(2,2)-TBINP(2,1)*TBINP(1,2)
        DISC=DSQRT(R**2-4.0*U*SAL)
        YO1=(-R+DISC)/(2.0*U)
        YO2=(-R-DISC)/(2.0*U)
       GO TO 69
   70  CONTINUE
        C=F/E
        DAO=G/E
```

```
      U=AB*(C**2+1.0)
      R=AB*(2.0*C*DAO-C*(TBINP(1,2)+TBINP(2,2))-TBINP(1,1)-TBINP(2,1))
     1   +C*(TBINP(2,1)-TBINP(1,1))+TBINP(1,2)-TBINP(2,2)
      SAL=AB*(DAO**2-DAO*(TBINP(1,2)+TBINP(2,2))+TBINP(1,1)*TBINP(2,1)
     2       +TBINP(1,2)*TBINP(2,2))+DAO*(TBINP(2,1)-TBINP(1,1))+
     3        TBINP(1,1)*TBINP(2,2)-TBINP(2,1)*TBINP(1,2)
      DISC=DSQRT(R**2-4.0*U*SAL)
      XO1=(-R+DISC)/(2.0*U)
      XO2=(-R-DISC)/(2.0*U)
      YO1=C*XO1+DAO
      YO2=C*XO2+DAO
69    CONTINUE
     GO TO 71
67   IF(TBINP(1,3).NE.(PI/2.0).OR.TBINP(2,3).EQ.(PI/2.0))GO TO 72
      BA=DTAN(TBINP(2,3))
      E=BA*(TBINP(3,2)-TBINP(1,2))+TBINP(2,1)-TBINP(3,1)
      F=BA*(TBINP(1,1)-TBINP(3,1))+TBINP(2,2)-TBINP(3,2)
      G=BA*(TBINP(2,2)*TBINP(3,2)+TBINP(2,1)*TBINP(3,1)-TBINP(1,2)*
     4        TBINP(2,2)-TBINP(1,1)*TBINP(2,1))+TBINP(2,1)*TBINP(3,2)-
     5        TBINP(3,1)*TBINP(2,2)
      IF(F.NE.0.0)GO TO 73
      DAO=G/E
      YO1=DAO
      YO2=DAO
      R=-TBINP(1,1)-TBINP(2,1)
      SAL=DAO**2-DAO*(TBINP(1,2)+TBINP(2,2))+TBINP(1,2)*TBINP(2,2)+
     6                   TBINP(1,1)*TBINP(2,1)
      DISC=DSQRT(R**2-4.0*SAL)
      XO1=(-R+DISC)/2.0
      XO2=(-R-DISC)/2.0
      GO TO 74
73    IF(E.NE.0.0)GO TO 75
      H=(-G/F)
      XO1=H
      XO2=H
      R=-TBINP(1,2)-TBINP(2,2)
      SAL=H**2-H*(TBINP(1,1)+TBINP(2,1))+TBINP(1,1)*TBINP(2,1)+
     7       TBINP(1,2)*TBINP(2,2)
      DISC=DSQRT(R**2-4.0*SAL)
      YO1=(-R+DISC)/2.0
      YO2=(-R-DISC)/2.0
      GO TO 74
75    CONTINUE
      C=F/E
      DAO=G/E
      U=C**2+1.0
      R=2.0*C*DAO-C*(TBINP(1,2)+TBINP(2,2))-TBINP(1,1)-TBINP(2,1)
      SAL=DAO**2-DAO*(TBINP(1,2)+TBINP(2,2))+TBINP(1,2)*TBINP(2,2)+
```

```
   8                TBINP(1,1)*TBINP(2,1)
      DISC=DSQRT(R**2-4.0*U*SAL)
      XO1=(-R+DISC)/(2.0*U)
      XO2=(-R-DISC)/(2.0*U)
      YO1=C*XO1+DAO
      YO2=C*XO2+DAO
 74  CONTINUE
    GO TO 71
 72 IF(TBINP(1,3).EQ.(PI/2.0).OR.TBINP(2,3).NE.(PI/2.0))GO TO 76
     AB=DTAN(TBINP(1,3))
     E=AB*(TBINP(1,2)-TBINP(3,2))+TBINP(1,1)-TBINP(3,1)
     F=AB*(TBINP(3,1)-TBINP(1,1))+TBINP(1,2)-TBINP(2,2)
     G=AB*(TBINP(1,1)*TBINP(2,1)+TBINP(1,2)*TBINP(2,2)-TBINP(2,1)*
   9          TBINP(3,1)-TBINP(2,2)*TBINP(3,2))+TBINP(1,1)*TBINP(2,2)-
   1                TBINP(2,1)*TBINP(1,2)
     IF(F.NE.0.0)GO TO 77
      DAO=G/E
      YO1=DAO
      YO2=DAO
      R=-TBINP(2,1)-TBINP(3,1)
      SAL=DAO**2-DAO*(TBINP(2,2)+TBINP(3,2))+TBINP(2,2)*TBINP(3,2)+
   1                TBINP(2,1)*TBINP(3,1)
      DISC=DSQRT(R**2-4.0*SAL)
      XO1=(-R+DISC)/2.0
      XO2=(-R-DISC)/2.0
     GO TO 78
 77   IF(E.NE.0.0)GO TO 79
      H=(-G/F)
      XO1=H
      XO2=H
      R=-TBINP(2,2)-TBINP(3,2)
      SAL=H**2-H*(TBINP(2,1)+TBINP(3,1))+TBINP(2,1)*TBINP(3,1)+
   2              TBINP(2,2)*TBINP(3,2)
      DISC=DSQRT(R**2-4.0*SAL)
      YO1=(-R+DISC)/2.0
      YO2=(-R-DISC)/2.0
     GO TO 78
 79  CONTINUE
      C=F/E
      DAO=G/E
      U=C**2+1.0
      R=2.0*C*DAO-C*(TBINP(2,2)+TBINP(3,2))-TBINP(2,1)-TBINP(3,1)
      SAL=DAO**2-DAO*(TBINP(2,2)+TBINP(3,2))+TBINP(2,1)*TBINP(3,1)+
   3                         TBINP(2,2)*TBINP(3,2)
      DISC=DSQRT(R**2-4.0*U*SAL)
      XO1=(-R+DISC)/(2.0*U)
      XO2=(-R-DISC)/(2.0*U)
      YO1=C*XO1+DAO
```

```
      YO2=C*XO2+DAO
78    CONTINUE
      GO TO 71
76   CONTINUE
      E=TBINP(1,2)-TBINP(3,2)
      F=TBINP(3,1)-TBINP(1,1)
      G=TBINP(1,1)*TBINP(2,1)+TBINP(1,2)*TBINP(2,2)-TBINP(2,2)*
    4             TBINP(3,2)-TBINP(2,1)*TBINP(3,1)
      IF(F.NE.0.0)GO TO 80
       DAO=G/E
       YO1=DAO
       YO2=DAO
       XO1=TBINP(1,1)
       XO2=TBINP(1,1)
      GO TO 81
80    IF(E.NE.0.0)GO TO 82
       H=(-G/F)
       XO1=H
       XO2=H
       YO1=TBINP(1,2)
       YO2=TBINP(1,2)
      GO TO 81
82    CONTINUE
       C=F/E
       DAO=G/E
       U=C**2+1.0
       R=2.0*C*DAO-C*(TBINP(2,2)+TBINP(3,2))-TBINP(2,1)-TBINP(3,1)
       SAL=DAO**2-DAO*(TBINP(2,2)+TBINP(3,2))+TBINP(2,2)*TBINP(3,2)+
    7                                            TBINP(2,1)*TBINP(3,1)
       DISC=DSQRT(R**2-4.0*U*SAL)
       XO1=(-R+DISC)/(2.0*U)
       XO2=(-R-DISC)/(2.0*U)
       YO1=C*XO1+DAO
       YO2=C*XO2+DAO
81    CONTINUE
71   CONTINUE
C     ALGORITHM SELECTION(TBINP,XO1,YO1,XO2,YO2)
       IF(TBINP(1,3).EQ.(PI/2.0))GO TO 83
       VALOR1=((TBINP(2,1)-XO1)*(TBINP(1,2)-YO1)-(TBINP(1,1)-XO1)*
    8           (TBINP(2,2)-YO1))/((TBINP(2,2)-YO1)*(TBINP(1,2)-YO1)+
    9           (TBINP(2,1)-XO1)*(TBINP(1,1)-XO1))
       VALOR2=((TBINP(2,1)-XO2)*(TBINP(1,2)-YO2)-(TBINP(1,1)-XO2)*
    1           (TBINP(2,2)-YO2))/((TBINP(2,2)-YO2)*(TBINP(1,2)-YO2)+
    1           (TBINP(2,1)-XO2)*(TBINP(1,1)-XO2))
       MODUL1=DABS(DTAN(TBINP(1,3))-VALOR1)
       MODUL2=DABS(DTAN(TBINP(1,3))-VALOR2)
       IF(MODUL1.GE.MODUL2)GO TO 84
         XO=XO1
```

```
         YO=YO1
        GO TO 85
 84     CONTINUE
         XO=XO2
         YO=YO2
 85    CONTINUE
      GO TO 86
 83   IF(TBINP(2,3).EQ.(PI/2.0))GO TO 87
       VALOR1=((TBINP(3,1)-XO1)*(TBINP(2,2)-YO1)-(TBINP(2,1)-XO1)*
     2            (TBINP(3,2)-YO1))/((TBINP(3,2)-YO1)*(TBINP(2,2)-YO1)+
     3            (TBINP(3,1)-XO1)*(TBINP(2,1)-XO1))
       VALOR2=((TBINP(3,1)-XO2)*(TBINP(2,2)-YO2)-(TBINP(2,1)-XO2)*
     4            (TBINP(3,2)-YO2))/((TBINP(3,2)-YO2)*(TBINP(2,2)-YO2)+
     5            (TBINP(3,1)-XO2)*(TBINP(2,1)-XO2))
       MODUL1=DABS(DTAN(TBINP(2,3))-VALOR1)
       MODUL2=DABS(DTAN(TBINP(2,3))-VALOR2)
       IF(MODUL1.GE.MODUL2)GO TO 88
        XO=XO1
        YO=YO1
       GO TO 89
 88    CONTINUE
        XO=XO2
        YO=YO2
 89    CONTINUE
      GO TO 86
 87   CONTINUE
       VALOR1=(TBINP(2,2)-YO1)*(TBINP(1,2)-YO1)+(TBINP(2,1)-XO1)*
     6                                            (TBINP(1,1)-XO1)
       VALOR2=(TBINP(2,2)-YO2)*(TBINP(1,2)-YO2)+(TBINP(2,1)-XO2)*
     7                                            (TBINP(1,1)-XO2)
       MODUL1=DABS(VALOR1)
       MODUL2=DABS(VALOR2)
       IF(MODUL1.GE.MODUL2)GO TO 90
        XO=XO1
        YO=YO1
       GO TO 91
 90    CONTINUE
        XO=XO2
        YO=YO2
 91    CONTINUE
 86   CONTINUE
C    END SELECTION(XO,YO)
C END INITIAL_COORDINATES(XO,YO)
C ALGORITHM RESTORE_INITIAL_DATA(TBINP,STORE)
     TBINP(1,1)=STORE(1)
     TBINP(1,2)=STORE(2)
     TBINP(1,3)=STORE(3)
     TBINP(2,1)=STORE(4)
```

```
      TBINP(2,2)=STORE(5)
      TBINP(2,3)=STORE(6)
      TBINP(3,1)=STORE(7)
      TBINP(3,2)=STORE(8)
      TBINP(I,1)=STORE(9)
      TBINP(I,2)=STORE(10)
      TBINP(I,3)=STORE(11)
      TBINP(J,1)=STORE(12)
      TBINP(J,2)=STORE(13)
      TBINP(J,3)=STORE(14)
      TBINP(K,1)=STORE(15)
      TBINP(K,2)=STORE(16)
C END RESTORE_INITIAL_DATA(TBINP)
C END FIRST_INITIAL_POINT_FOR_FIX_BY_N_SEXTANT_ANGLES(XO,YO)
C ALGORITHM ITERATIONS(TOL)
   51 CONTINUE
C ALGORITHM INITIAL_AZIMUTHS(XO,YO,TBINP,M)
      DO 92 I=1,M
       IF(YO.NE.TBINP(I,2).OR.XO.LE.TBINP(I,1))GO TO 93
        AZ(I)=(3.0*PI)/2.0
       GO TO 94
   93  IF(YO.NE.TBINP(I,2).OR.XO.GE.TBINP(I,1))GO TO 95
        AZ(I)=PI/2.0
       GO TO 94
   95  CONTINUE
        ALFA(I)=DATAN((TBINP(I,1)-XO)/(TBINP(I,2)-YO))
        IF(ALFA(I).LT.0.0.OR.XO.GE.TBINP(I,1))GO TO 96
         AZ(I)=ALFA(I)
        GO TO 97
   96   IF(ALFA(I).GE.0.0.OR.XO.LE.TBINP(I,1))GO TO 98
         AZ(I)=ALFA(I)+2.0*PI
        GO TO 97
   98   CONTINUE
         AZ(I)=ALFA(I)+PI
   97   CONTINUE
   94  CONTINUE
   92 CONTINUE
C END INITIAL_AZIMUTHS(AZ)
C ALGORITHM MATRIX_L(TBINP(I,3),AZ,N)
C     ALGORITHM ZERO(L)
       DO 99 I=1,10
        L(I)=0.000000000
   99  CONTINUE
C     END ZERO(L)
      DO 210 I=1,N
       J=I+1
       L(I)=TBINP(I,3)+AZ(I)-AZ(J)
  210 CONTINUE
```

```
C END MATRIX_L(L)
C ALGORITHM SQUARED DISTANCES(TBINP,XO,YO)
      DO 211 I=1,M
       SO(I)=(TBINP(I,1)-XO)**2+(TBINP(I,2)-YO)**2
  211 CONTINUE
C END SQUARED_DISTANCES(SO)
C ALGORITHM MATRIX_A(N,TBINP,XO,YO,SO)
C     ALGORITHM ZERO(A)
      DO 212 I=1,10
      DO 213 J=1,2
         A(I,J)=0.000000000
  213   CONTINUE
  212  CONTINUE
C     END ZERO(A)
      DO 214 I=1,N
       J=I+1
       A(I,1)=(YO-TBINP(J,2))/SO(J)-(YO-TBINP(I,2))/SO(I)
  214 CONTINUE
      DO 215 I=1,N
       J=I+1
       A(I,2)=(XO-TBINP(I,1))/SO(I)-(XO-TBINP(J,1))/SO(J)
  215 CONTINUE
C END MATRIX_A(A)
C ALGORITHM NORMAL_EQUATIONS(A,W,L)
C ALGORITHM TRANSPOSE(A)*TBW(A,TBW)
      DO 43 I=1,2
       DO 44 J=1,10
        ATW(I,J)=.0000000000
        DO 45 K=1,10
         ATW(I,J)=ATW(I,J)+A(K,I)*TBW(K,J)
   45   CONTINUE
   44  CONTINUE
   43 CONTINUE
C END TRANSPOSE(A)*TBW(ATW)
C ALGORITHM MATRIX_ATWA(ATW,A)
      DO 46 I=1,2
       DO 47 J=1,2
        ATWA(I,J)=.0000000000
        DO 48 K=1,10
         ATWA(I,J)=ATWA(I,J)+ATW(I,K)*A(K,J)
   48   CONTINUE
   47  CONTINUE
   46 CONTINUE
C END MATRIX_ATWA(ATWA)
C ALGORITHM INVERT_ATWA(ATWA)
      BETA=ATWA(1,2)**2-ATWA(1,1)*ATWA(2,2)
      Q(1,1)=-ATWA(2,2)/BETA
      Q(1,2)=ATWA(1,2)/BETA
```

```
      Q(2,1)=Q(1,2)
      Q(2,2)=-ATWA(1,1)/BETA
C END INVERT_ATWA(Q)
C ALGORITHM MATRIX_ATWL(ATWL)
      DO 49 I=1,2
       ATWL(I)=.0000000000
       DO 50 K=1,10
        ATWL(I)=ATWL(I)+ATW(I,K)*L(K)
   50  CONTINUE
   49 CONTINUE
C END MATRIX_ATWL(ATWL)
C ALGORITHM ADJUSTED_INCREMENTS(Q,ATWL)
      DELTAX=Q(1,1)*ATWL(1)+Q(1,2)*ATWL(2)
      DELTAY=Q(2,1)*ATWL(1)+Q(2,2)*ATWL(2)
C END ADJUSTED_INCREMENTS(DELTAX,DELTAY)
C END NORMAL_EQUATIONS(DELTAX,DELTAY)
C ALGORITHM NEW_INITIAL_POINT(XO,YO,DELTAX,DELTAY)
      XO=XO+DELTAX
      YO=YO+DELTAY
C END NEW_INITIAL_POINT(XO,YO)
C ALGORITHM TOLERANCE(DELTAX,DELTAY)
      TOL=DELTAX**2+DELTAY**2
C END TOLERANCE(TOL)
      IF(TOL.GE.1.000)GO TO 51
C END ITERATIONS(XO,YO)
C ALGORITHM FINAL_ADJUSTED_POSITION(XO,YO)
      WRITE(6,126)XO,YO
C FINAL_ADJUSTED_POSITION(X,Y)
C ALGORITHM PRECISION(A,TBW,Q,L,DELTAX,DELTAY,N)
C ALGORITHM RESIDUALS(A,L,DELTAX,DELTAY,N)
      X(1)=DELTAX
      X(2)=DELTAY
C     ALGORITHM AX(A,X,N)
      DO 52 I=1,N
       AX(I)=.000000000
       DO 53 J=1,2
        AX(I)=AX(I)+A(I,J)*X(J)
   53  CONTINUE
   52 CONTINUE
C     END AX(AX)
C ALGORITHM V(AX,L,N)
      DO 54 I=1,N
       V(I)=AX(I)-L(I)
   54 CONTINUE
C     END V(V)
C     END RESIDUALS (V)
C ALGORITHM ST_DEVIATION_OF_UNIT_WEIGHT_OBS(V,TBW,N)
C     ALGORITHM VTW(V,W,N)
```

```
          DO 55  I=1,N
           VTW(I)=V(I)*TBW(I,I)
    55    CONTINUE
C        END VTW(VTW)
C        ALGORITHM VTWV(VTW,V)
          VTWV=.000000000
          DO 56  I=1,N
           VTWV=VTWV+VTW(I)*V(I)
    56    CONTINUE
C        END VTWV(VTWV)
C        ALGORITHM TRACE(TBW)
          TRACE=.0000000000
          DO 57  I=1,N
           TRACE=TRACE+TBW(I,I)
    57    CONTINUE
C         END TRACE(TRACE)
C         ALGORITHM SO(VTWV,TRACE)
          CHARLE=VTWV/(TRACE-2.000000000)
          SU=DSQRT(CHARLE)
C        END SO(SO)
C END ST_DEVIATION_OF_UNIT_WEIGHT_OBS(SO)
C ALGORITHM ST_DEVIATION_OF_EACH_OBS(SO,TBW)
        WRITE(6,134)
        DO 58 I=1,N
         S=(SU/DSQRT(TBW(I,I)))*(180.000000000/PI)
         WRITE(6,135)I,S
    58 CONTINUE
C END ST_DEVIATION_OF_EACH_OBS
C ALGORITHM ST_DEVIATIONS_AND_COVARIANCE_OF_X_AND_Y(SO,Q)
        SX=SU*DSQRT(Q(1,1))
        SY=SU*DSQRT(Q(2,2))
        SXY=(SU**2)*Q(1,2)
        WRITE(6,136)SX,SY,SXY
C END ST_DEVIATIONS_AND_COVARIANCE_OF_X_AND_Y(SX,SY,SXY)
C ALGORITHM CORRELATION_CEFFICIENT(SX,SY,SXY)
        RO=SXY/(SX*SY)
        WRITE(6,137)RO
C END CORRELATION_COEFFICIENT(RO)
C END PRECISION(SO,SX,SY,SXY,RO)
C ALGORITHM ERROR_ELIPSE(Q,SU)
        WRITE(6,138)
C ALGORITHM D(Q)
        D=DSQRT((Q(1,1)-Q(2,2))**2+4.000000000*(Q(1,2)**2))
C END D(D)
C ALGORITHM SEMI-MAJOR AXIS(SU,Q)
        SA=SU*DSQRT(2.000000000*Q(1,1)*Q(2,2)/(Q(1,1)+Q(2,2)-D))
        WRITE(6,139)SA
C END SEMI-MAJOR AXIS(SA)
```

```
C ALGORITHM SEMI-MINOR AXIS(SU,Q)
      SB=SU*DSQRT(2.00000000*Q(1,1)*Q(2,2)/(Q(1,1)+Q(2,2)+D))
      WRITE(6,140)SB
C END SEMI-MINOR AXIS(SB)
C ALGORITHM GAMA(Q)
      IF(Q(1,1).NE.Q(2,2))GO TO 59
        GAMA=PI/4.00000000000
      GO TO 60
   59  CONTINUE
        OMEGA=DATAN(2.00000000*Q(1,2)/(Q(1,1)-Q(2,2)))
        IF(OMEGA.LT.0.000)GO TO 61
         GAMA=OMEGA/2.000000000
        GO TO 62
   61   CONTINUE
          GAMA=(OMEGA+PI)/2.00000000
   62   CONTINUE
   60 CONTINUE
C END GAMA(GAMA)
C ALGORITHM INTERSECTION(SU,Q,GAMA)
      X10=(SU**2)*Q(1,1)*Q(2,2)
      X11=Q(2,2)-2.00000000*Q(1,2)*DTAN(GAMA)+(DTAN(GAMA)**2)*Q(1,1)
      X1=X10/X11
      Y1=(DTAN(GAMA)**2)*X1
C END INTERSECTION(X1,Y1)
C ALGORITHM AVERAGE(SA,SB)
      AVER=((SA+SB)/2.00000000)**2
C END AVERAGE(AVER)
C ALGORITHM SELECTION(AVER,X1,Y1)
      D1=X1+Y1
      IF(D1.LT.AVER)GO TO 63
        GAMAO=GAMA
      GO TO 64
   63 CONTINUE
        GAMAO=GAMA+PI/2.00000000000
   64 CONTINUE
      GAMAO=GAMAO*(180.00000000/PI)
      WRITE(6,142)GAMAO
C END SELECTION(GAMAO)
C END ERROR ELIPSE(SA,SB,GAMAO)
  998 CONTINUE
      WRITE(6,191)
      DO 257 I=1,N
       READ(5,188)TBINP(I,3)
  257 CONTINUE
C     (INTRODUCE NEW DATA SET.IF NO MORE DATA IS AVAILABLE THEN THE LAST
C      SET IS WITH ALL SEXTANT ANGLES EQUAL TO 400.0)
      GO TO 255
  999 CONTINUE
```

```
C END FIX_BY_N_SEXTANT_ANGLES
      STOP
      END
```

```
// EXEC FORTXCLG
//FORT.SYSIN DD *
C PROGRAM FOR FIX DETERMINATION GIVEN TWO RANGE DISTANCES
C    AND ONE AZIMUTH ANGLE
C
C THIS PROGRAM DETERMINES ADJUSTED COORDINATES OF VESSEL,USING LEAST
C SQUARES METHOD.ALSO GIVES INFORMATION ABOUT PRECISION OF OBSERVED
C VALUES AND COMPUTED RESULTS,INCLUDING ERROR ELIPSE
C
C USER'S INSTRUCTIONS
C 1-ALWAYS CONSIDER 3 STATIONS(DESIGNATED BY S1,S2 AND S3)
C 2-IF THE DISTANCES AND AZIMUTH ARE OBSERVED FROM 3 DIFFERENT
C    STATIONS,THEN DESIGNATE BY S3 THE STATION FROM WHICH THE AZIMUTH
C    IS OBSERVED.THE REMAINING 2 STATIONS ARE,INDIFFERENTLY,DESIGNATED
C    BY S1 AND S2.
C 2-IF THE DISTANCES AND AZIMUTH ARE OBSERVED FROM JUST 2 DIFFERENT
C    STATIONS,STATION S3 IS THE STATION FROM WHICH THE AZIMUTH ANGLE
C    IS OBSERVED.
C    S1 COINCIDE WITH S3(A RANGE DISTANCE IS OBSERVED FROM S1)
C    THE REMAINING STATION IS DESIGNATED BY S2(FROM WHICH ANOTHER
C    RANGE DISTANCE IS OBSERVED)
C 3-INPUT FOR EACH STATION  THE X-COORDINATE(EASTING),Y-COORDINATE
C    (NORTHING) AND STANDARD ERROR OF OBSERVED VALUE USING FORMAT 160
C    IF THERE IS NO INFORMATION ABOUT STANDARD ERROR,ENTER 1.0
C 4-INPUT THE RANGE DISTANCES AND AZIMUTH ANGLE OBSERVED FROM
C    S1,S2 AND S3 USING SAME FORMAT 160(ONE CARD FOR EACH DATA SET)
C 5-WHEN NO MORE DATA IS AVAILABLE,INPUT A 'DUMMY'DATA SET WITH
C    ALL VALUES EQUAL TO 0.0 USING SAME FORMAT 160
C
C
C ALGORITHM FIX_BY_TWO_RANGES_AND_ONE_AZIMUTH
      INTEGER N,I,J,K
      REAL*8 PI,TBINP(10,4),E,E1,E2,E3,E4,X0,Y0,X01,Y01,X02,Y02,A301,
     1       A302,F,DSQRT,A30,S10,S20,S30,A(10,2),DSIN,DCOS,L(10),
     2       X(2),AX(10),V(10),VTW(10),VTWV,TRACE,CHARLE,SU,SX,SY,SXY,
     3       RO,D,SA,SB,GAMA,OMEGA,X10,X11,X1,Y1,AVER,D1,GAMAO,DARSIN,
     4       S,TBW(10,10),GREAT,ATW(2,10),ATWA(2,2),Q(2,2),BETA,ATWL(2),
     5       DELTAX,DELTAY,TOL,DTAN,DATAN
  126 FORMAT(//////,1X,'ADJUSTED COORDINATES X=',F13.3,3X,'Y=',F13.3)
  136 FORMAT('0',1X,'SX=',F6.2,3X,'SY=',F6.2,3X,'SXY=',F9.3)
  137 FORMAT('0',1X,'CORRELATION COEFFICIENT RO=',F5.2)
  138 FORMAT(//////,2X,'ERROR ELIPSE SEMI-AXIS AND ORIENTATION')
  139 FORMAT(/,2X,'SEMI-MAJOR AXIS SA=',F6.3)
  140 FORMAT(/,2X,'SEMI-MINOR AXIS SB=',F6.3)
  142 FORMAT(/,2X,'ANGLE FROM X-AXIS TO SA ANTICLOCKWISE=',F5.1,'DEG')
  160 FORMAT(1X,2F12.2,F7.3)
  161 FORMAT('0',1X,'ST#',I2,3X,'EAST=',F12.2,3X,'NORT=',F12.2,3X,
     1       'ST ERROR=',F7.3,2X,'METERS')
```

```
  162 FORMAT('0',1X,'ST# 3   EAST=',F12.2,3X,'NORT=',F12.2,3X,
     2       'ST ERROR=',F7.3,2X,'DEGREES')
  163 FORMAT('1',1X)
  164 FORMAT('0',1X,'R1=',F9.2,2X,'METERS')
  165 FORMAT('0',1X,'R2=',F9.2,2X,'METERS')
  166 FORMAT('0',1X,'A=',F10.2,2X,'DEGREES')
  167 FORMAT(//////,1X,'SOLUTION UNDETERMINED FOR THAT DATA SET')
  175 FORMAT(//////,1X,'PRECISION OF OBSERVATIONS')
  176 FORMAT('0',1X,'ST DEVIATION OF OBS',I2,'  =',F6.2,'  METERS')
  177 FORMAT('0',1X,'ST DEVIATION OF OBS 3  =',F6.3,'  DEGREES')
  193 FORMAT(//////,1X,'OBSERVED RANGE DISTANCES AND AZIMUTH ANGLE')
C (CONSIDER ALWAYS 3 STATIONS.THE THIRD STATION WILL BE THE ONE FROM
C  WICH AN AZIMUTH IS OBSERVED.FOR EACH STATION INPUT X-COORDINATE,
C  Y-COORDINATE,AND STANDARD ERROR)
      N=3
      PI=3.141592653589793
      WRITE(6,163)
      DO 218 I=1,3
       READ(5,160)TBINP(I,1),TBINP(I,2),TBINP(I,4)
  218 CONTINUE
      DO 219 I=1,2
       WRITE(6,161)I,TBINP(I,1),TBINP(I,2),TBINP(I,4)
  219 CONTINUE
      WRITE(6,162)TBINP(3,1),TBINP(3,2),TBINP(3,4)
C     ALGORITHM CONVERSION DEGREES RADIANS(TBINP(3,4))
       TBINP(3,4)=TBINP(3,4)*(PI/180.0)
C     END CONVERSION DEGREES RADIANS(TBINP(3,4))
      READ(5,160)TBINP(1,3),TBINP(2,3),TBINP(3,3)
C       (INPUT OBSERVED RANGES R1 AND R2 IN METERS AND OBSERVED
C        AZIMUTH A IN DEGREES)
  220 IF(TBINP(1,3).EQ.0.0)GO TO 999
       WRITE(6,193)
       WRITE(6,164)TBINP(1,3)
       WRITE(6,165)TBINP(2,3)
       WRITE(6,166)TBINP(3,3)
C      ALGORITHM CONVERSION DEGREES RADIANS(TBINP(3,3))
        TBINP(3,3)=TBINP(3,3)*(PI/180.0)
C      END CONVERSION DEGREES RADIANS(TBINP(3,3))
C ALGORITHM FIRST INITIAL POINT(TBINP)
       E=TBINP(1,3)**2-TBINP(2,3)**2+TBINP(2,2)**2-TBINP(1,2)**2+
     1   TBINP(2,1)**2-TBINP(1,1)**2
      IF(TBINP(2,1).EQ.TBINP(1,1))GO TO 227
       E1=((TBINP(1,2)-TBINP(2,2))/(TBINP(2,1)-TBINP(1,1)))**2+1.0
       E2=(E*(TBINP(1,2)-TBINP(2,2)))/((TBINP(2,1)-TBINP(1,1))**2)-
     2    2.0*TBINP(1,1)*((TBINP(1,2)-TBINP(2,2))/
     3    (TBINP(2,1)-TBINP(1,1)))-2.0*TBINP(1,2)
       E3=(E/(2.0*(TBINP(2,1)-TBINP(1,1))))**2-(E*TBINP(1,1))/
     +    (TBINP(2,1)-TBINP(1,1))-TBINP(1,3)**2+TBINP(1,1)**2+
```

```
     5      TBINP(1,2)**2
          E4=E2**2-4.0*E1*E3
          IF(E4.GE.0.0)GO TO 228
           XO=TBINP(1,1)+TBINP(1,3)*(TBINP(2,1)-TBINP(1,1))/DSQRT
     6          ((TBINP(2,1)-TBINP(1,1))**2+(TBINP(2,2)-TBINP(1,2))**2)
           YO=TBINP(1,2)+TBINP(1,3)*(TBINP(2,2)-TBINP(1,2))/DSQRT
     7          ((TBINP(2,1)-TBINP(1,1))**2+(TBINP(2,2)-TBINP(1,2))**2)
          GO TO 229
   228    IF(E4.NE.0.0)GO TO 230
           YO=E2/(-2.0*E1)
           XO=E/(2.0*(TBINP(2,1)-TBINP(1,1)))+YO*(TBINP(1,2)-TBINP(2,2))/
     8          (TBINP(2,1)-TBINP(1,1))
          GO TO 229
   230    CONTINUE
           YO1=(-E2+DSQRT(E4))/(2.0*E1)
           XO1=E/(2.0*(TBINP(2,1)-TBINP(1,1)))+YO1*(TBINP(1,2)-TBINP(2,2))/
     1          (TBINP(2,1)-TBINP(1,1))
           YO2=(-E2-DSQRT(E4))/(2.0*E1)
           XO2=E/(2.0*(TBINP(2,1)-TBINP(1,1)))+YO2*(TBINP(1,2)-TBINP(2,2))/
     2          (TBINP(2,1)-TBINP(1,1))
           IF(TBINP(3,1).NE.XO1.OR.TBINP(3,2).NE.YO1)GO TO 231
            XO=XO2
            YO=YO2
           GO TO 232
   231     IF(TBINP(3,1).NE.XO2.OR.TBINP(3,2).NE.YO2)GO TO 233
            XO=XO1
            YO=YO1
           GO TO 232
   233     CONTINUE
            CALL CRITER(TBINP(3,1),TBINP(3,2),XO1,YO1,A301)
            CALL CRITER(TBINP(3,1),TBINP(3,2),XO2,YO2,A302)
            IF(A301.NE.TBINP(3,3).OR.A301.NE.A302)GO TO 234
             WRITE(6,167)
            GO TO 993
   234      IF(((A301-TBINP(3,3))**2).NE.((A302-TBINP(3,3))**2))GO TO 235
             XO=(XO1+XO2)/2.0
             YO=(YO1+YO2)/2.0
            GO TO 236
   235      IF(((A301-TBINP(3,3))**2).LE.((A302-TBINP(3,3))**2))GO TO 237
             XO=XO2
             YO=YO2
            GO TO 236
   237      IF(((A301-TBINP(3,3))**2).GE.((A302-TBINP(3,3))**2))GO TO 236
             XO=XO1
             YO=YO1
   236      CONTINUE
   232     CONTINUE
   229    CONTINUE
```

```
      GO TO 238
  227 CONTINUE
       YO=E/(2.0*(TBINP(2,2)-TBINP(1,2)))
       F=TBINP(1,3)**2-(TBINP(1,2)-YO)**2
       IF(F.GT.0.0)GO TO 239
        XO=TBINP(1,1)
       GO TO 240
  239  CONTINUE
        XO1=TBINP(1,1)+DSQRT(F)
        YO1=YO
        XO2=TBINP(1,1)-DSQRT(F)
        YO2=YO
        IF(TBINP(3,1).NE.XO1.OR.TBINP(3,2).NE.YO1)GO TO 241
         XO=XO2
        GO TO 242
  241   IF(TBINP(3,1).NE.XO2.OR.TBINP(3,2).NE.YO2)GO TO 243
         XO=XO1
        GO TO 242
  243   CONTINUE
         CALL CRITER(TBINP(3,1),TBINP(3,2),XO1,YO1,A301)
         CALL CRITER(TBINP(3,1),TBINP(3,2),XO2,YO2,A302)
         IF(A301.NE.TBINP(3,3).OR.A301.NE.A302)GO TO 244
          WRITE(6,167)
         GO TO 998
  244    IF(((A301-TBINP(3,3))**2).NE.((A302-TBINP(3,3))**2))GO TO 245
          XO=XO1
         GO TO 246
  245    IF(((A301-TBINP(3,3))**2).LE.((A302-TBINP(3,3))**2))GO TO 247
          XO=XO2
         GO TO 246
  247    IF(((A301-TBINP(3,3))**2).GE.((A302-TBINP(3,3))**2))GO TO 246
          XO=XO1
  246    CONTINUE
  242   CONTINUE
  240  CONTINUE
  238 CONTINUE
C END FIRST_INITIAL_POINT(XO,YO)
C ALGORITHM ITERATIONS(TOL)
   51 CONTINUE
C ALGORITHM A30(TBINP,XO,YO)
      CALL CRITER(TBINP(3,1),TBINP(3,2),XO,YO,A30)
C END A30(A30)
C ALGORITHM DISTANCES(TBINP,XO,YO)
      S10=DSQRT((TBINP(1,1)-XO)**2+(TBINP(1,2)-YO)**2)
      S20=DSQRT((TBINP(2,1)-XO)**2+(TBINP(2,2)-YO)**2)
      S30=DSQRT((TBINP(3,1)-XO)**2+(TBINP(3,2)-YO)**2)
C END DISTANCES(S10,S20,S30)
C ALGORITHM MATRIX_A(TBINP,XO,YO,S10,S20,S30,A30)
```

```
C        ALGORITHM ZERO(A)
         DO 38 I=1,10
          DO 39 J=1,2
           A(I,J)=0.0000000000
   39     CONTINUE
   38    CONTINUE
C       END ZERO(A)
        A(1,1)=(XO-TBINP(1,1))/S10
        A(1,2)=(YO-TBINP(1,2))/S10
        A(2,1)=(XO-TBINP(2,1))/S20
        A(2,2)=(YO-TBINP(2,2))/S20
        A(3,1)=DCOS(A30-TBINP(3,3))*(YO-TBINP(3,2))/S30+
     1          DSIN(A30-TBINP(3,3))*(XO-TBINP(3,1))/S30
        A(3,2)=DCOS(A30-TBINP(3,3))*(TBINP(3,1)-XO)/S30+
     2          DSIN(A30-TBINP(3,3))*(YO-TBINP(3,2))/S30
C END MATRIX_A(A)
C ALGORITHM LIST_L(TBINP,S10,S20,S30,A30)
C       ALGORITHM ZERO(L)
         DO 34 I=1,10
          L(I)=0.000000000
   34    CONTINUE
C       END ZERO(L)
        L(1)=TBINP(1,3)-S10
        L(2)=TBINP(2,3)-S20
        L(3)=DSIN(TBINP(3,3)-A30)*S30
C END LIST_L(L)
C ALGORITHM BEFORE_WEIGHT_MATRIX(TBINP(3,4),S30)
        TBINP(3,4)=S30*DSIN(TBINP(3,4))
C END BEFORE_WEIGTH_MATRIX(TBINP(3,4))
C ALGORITHM WEIGHT_MATRIX(N,TBINP(I,4))
C       ALGORITHM ZERO(TBW)
         DO 11 I=1,10
          DO 12 J=1,10
           TBW(I,J)=.000000000
   12     CONTINUE
   11    CONTINUE
C       END ZERO(TBW)
C       ALGORITHM SQUARE(N,TBINP(I,4),TBW)
         DO 13 I=1,N
          TBW(I,I)=TBINP(I,4)**2
   13    CONTINUE
C       END SQUARE(TBW)
C       ALGORITHM NORMALIZE(TBW)
         GREAT=TBW(1,1)
         DO 14 I=2,N
          IF(TBW(I,I).GT.GREAT)GREAT=TBW(I,I)
   14    CONTINUE
         DO 15 I=1,N
```

```
          TBW(I,I)=GREAT/TBW(I,I)
   15   CONTINUE
C       END NORMALIZE(TBW)
C  END WEIGHT_MATRIX(TBW)
C  ALGORITHM AFTER_WEIGHT_MATRIX(TBINP(3,4),S30)
        TBINP(3,4)=DARSIN(TBINP(3,4)/S30)
C  AFTER_WEIGHT_MATRIX
C    ALGORITHM NORMAL_EQUATIONS(A,TBW,L)
C  ALGORITHM TRANSPOSE(A)*TBW(A,TBW)
        DO 43 I=1,2
          DO 44 J=1,10
            ATW(I,J)=.0000000000
            DO 45 K=1,10
              ATW(I,J)=ATW(I,J)+A(K,I)*TBW(K,J)
   45       CONTINUE
   44     CONTINUE
   43   CONTINUE
C  END TRANSPOSE(A)*TBW(ATW)
C  ALGORITHM MATRIX_ATWA(ATW,A)
        DO 46 I=1,2
          DO 47 J=1,2
            ATWA(I,J)=.0000000000
            DO 48 K=1,10
              ATWA(I,J)=ATWA(I,J)+ATW(I,K)*A(K,J)
   48       CONTINUE
   47     CONTINUE
   46   CONTINUE
C  END MATRIX_ATWA(ATWA)
C  ALGORITHM INVERT_ATWA(ATWA)
        BETA=ATWA(1,2)**2-ATWA(1,1)*ATWA(2,2)
        Q(1,1)=-ATWA(2,2)/BETA
        Q(1,2)=ATWA(1,2)/BETA
        Q(2,1)=Q(1,2)
        Q(2,2)=-ATWA(1,1)/BETA
C  END INVERT_ATWA(Q)
C  ALGORITHM MATRIX_ATWL(ATWL)
        DO 49 I=1,2
          ATWL(I)=.0000000000
          DO 50 K=1,10
            ATWL(I)=ATWL(I)+ATW(I,K)*L(K)
   50     CONTINUE
   49   CONTINUE
C  END MATRIX_ATWL(ATWL)
C  ALGORITHM ADJUSTED_INCREMENTS(Q,ATWL)
        DELTAX=Q(1,1)*ATWL(1)+Q(1,2)*ATWL(2)
        DELTAY=Q(2,1)*ATWL(1)+Q(2,2)*ATWL(2)
C  END ADJUSTED_INCREMENTS(DELTAX,DELTAY)
C    END NORMAL_EQUATIONS(DELTAX,DELTAY)
```

```
C ALGORITHM NEW_INITIAL_POINT(XO,YO,DELTAX,DELTAY)
      XO=XO+DELTAX
      YO=YO+DELTAY
C END NEW_INITIAL_POINT(XO,YO)
C ALGORITHM TOLERANCE(DELTAX,DELTAY)
      TOL=DELTAX**2+DELTAY**2
C END TOLERANCE(TOL)
      IF(TOL.GE.1.0000)GO TO 51
C END ITERATIONS(XO,YO)
C ALGORITHM FINAL_ADJUSTED_POSITION(XO,YO)
      WRITE(6,126)XO,YO
C FINAL_ADJUSTED_POSITION(X,Y)
C ALGORITHM PRECISION(A,TBW,Q,L,DELTAX,DELTAY,N,S30)
C ALGORITHM RESIDUALS(A,L,DELTAX,DELTAY,N)
      X(1)=DELTAX
      X(2)=DELTAY
C     ALGORITHM AX(A,X,N)
      DO 52 I=1,N
       AX(I)=.000000000
       DO 53 J=1,2
        AX(I)=AX(I)+A(I,J)*X(J)
   53  CONTINUE
   52 CONTINUE
C     END AX(AX)
C ALGORITHM V(AX,L,N)
      DO 54 I=1,N
       V(I)=AX(I)-L(I)
   54 CONTINUE
C     END V(V)
C     END RESIDUALS (V)
C ALGORITHM ST_DEVIATION_OF_UNIT_WEIGHT_OBS(V,TBW,N)
C     ALGORITHM VTW(V,W,N)
       DO 55 I=1,N
        VTW(I)=V(I)*TBW(I,I)
   55  CONTINUE
C     END VTW(VTW)
C     ALGORITHM VTWV(VTW,V)
       VTWV=.000000000
       DO 56 I=1,N
        VTWV=VTWV+VTW(I)*V(I)
   56  CONTINUE
C     END VTWV(VTWV)
C     ALGORITHM TRACE(TBW)
       TRACE=.0000000000
       DO 57 I=1,N
        TRACE=TRACE+TBW(I,I)
   57  CONTINUE
C      END TRACE(TRACE)
```

```
C       ALGORITHM SO(VTWV,TRACE)
        CHARLE=VTWV/(TRACE-2.000000000)
        SU=DSQRT(CHARLE)
C      END SO(SO)
C END ST_DEVIATION_OF_UNIT_WEIGHT_OBS(SO)
C ALGORITHM ST_DEVIATION_OF_EACH_OBS(SU,TBW,S30)
       WRITE(6,175)
       DO 248 I=1,2
        S=(SU/DSQRT(TBW(I,I)))
        WRITE(6,176)I,S
  248  CONTINUE
       S=DARSIN(SU/(DSQRT(TBW(3,3))*S30))*(180.0/PI)
       WRITE(6,177)S
C END ST_DEVIATION_OF_EACH_OBS
C ALGORITHM ST_DEVIATIONS_AND_COVARIANCE_OF_X_AND_Y(SO,Q)
       SX=SU*DSQRT(Q(1,1))
       SY=SU*DSQRT(Q(2,2))
       SXY=(SU**2)*Q(1,2)
       WRITE(6,136)SX,SY,SXY
C END ST_DEVIATIONS_AND_COVARIANCE_OF_X_AND_Y(SX,SY,SXY)
C ALGORITHM CORRELATION_CEFFICIENT(SX,SY,SXY)
       RO=SXY/(SX*SY)
       WRITE(6,137)RO
C END CORRELATION_COEFFICIENT(RO)
C END PRECISION(SU,SX,SY,SXY,RO)
C ALGORITHM ERROR_ELIPSE(Q,SU)
       WRITE(6,138)
C ALGORITHM D(Q)
       D=DSQRT((Q(1,1)-Q(2,2))**2+4.00000000*(Q(1,2)**2))
C END D(D)
C ALGORITHM SEMI-MAJOR AXIS(SU,Q)
       SA=SU*DSQRT(2.00000000*Q(1,1)*Q(2,2)/(Q(1,1)+Q(2,2)-D))
       WRITE(6,139)SA
C END SEMI-MAJOR AXIS(SA)
C ALGORITHM SEMI-MINOR AXIS(SU,Q)
       SB=SU*DSQRT(2.00000000*Q(1,1)*Q(2,2)/(Q(1,1)+Q(2,2)+D))
       WRITE(6,140)SB
C END SEMI-MINOR AXIS(SB)
C ALGORITHM GAMA(Q)
       IF(Q(1,1).NE.Q(2,2))GO TO 59
         GAMA=PI/4.00000000000
       GO TO 60
   59   CONTINUE
         OMEGA=DATAN(2.00000000*Q(1,2)/(Q(1,1)-Q(2,2)))
         IF(OMEGA.LT.0.000)GO TO 61
          GAMA=OMEGA/2.000000000
         GO TO 62
   61    CONTINUE
```

```
            GAMA=(OMEGA+PI)/2.00000000
   62     CONTINUE
   60  CONTINUE
C END GAMA(GAMA)
C ALGORITHM INTERSECTION(SU,Q,GAMA)
       X10=(SU**2)*Q(1,1)*Q(2,2)
       X11=Q(2,2)-2.00000000*Q(1,2)*DTAN(GAMA)+(DTAN(GAMA)**2)*Q(1,1)
       X1=X10/X11
       Y1=(DTAN(GAMA)**2)*X1
C END INTERSECTION(X1,Y1)
C ALGORITHM AVERAGE(SA,SB)
       AVER=((SA+SB)/2.00000000)**2
C END AVERAGE(AVER)
C ALGORITHM SELECTION(AVER,X1,Y1)
       D1=X1+Y1
       IF(D1.LT.AVER)GO TO 63
          GAMAD=GAMA
       GO TO 64
   63  CONTINUE
          GAMAO=GAMA+PI/2.0000000000
   64  CONTINUE
       GAMAO=GAMAO*(180.00000000/PI)
       WRITE(6,142)GAMAO
C END SELECTION(GAMAO)
C END ERROR_ELIPSE(SA,SB,GAMAO)
  998   CONTINUE
        READ(5,160)TBINP(1,3),TBINP(2,3),TBINP(3,3)
       WRITE(6,163)
C          (INTRODUCE NEW DATA SET.IF NO MORE DATA IS AVAILABLE,THE
C           LAST SET IS R1=0.0,R2=0.0,A=0.0)
       GO TO 220
  999  CONTINUE
C END FIX_BY_TWO_RANGES_AND_ONE_AZIMUTH
       STOP
       END
       SUBROUTINE CRITER(XS,YS,XP,YP,ASP)
        REAL*8 XS,YS,XP,YP,ASP,PI,ALFA,DATAN
        PI=3.141592653589793
        IF(YP.NE.YS.OR.XP.LE.XS)GO TO 221
         ASP=PI/2.0
        GO TO 222
  221   IF(YP.NE.YS.OR.XP.GE.XS)GO TO 223
         ASP=3.0*PI/2.0
        GO TO 222
  223   CONTINUE
         ALFA=DATAN((XP-XS)/(YP-YS))
         IF(ALFA.LT.0.0.OR.XP.LT.XS)GO TO 224
          ASP=ALFA
```

```
      GO TO 225
224   IF(ALFA.GE.0.0.OR.XP.GE.XS)GO TO 226
       ASP=ALFA+2.0*PI
      GO TO 225
226   CONTINUE
       ASP=ALFA+PI
225   CONTINUE
222  CONTINUE
     RETURN
    END
```

LIST OF REFERENCES

1. Ewing, C. E. and Mitchell, M. M., Introduction to Geodesy, Elsevier, 1976.

2. Wolf, P. R., Elements of Photogrammetry, p. 513, McGraw-Hill, 1974.

BIBLIOGRAPHY

Freund and Walpole, Mathematical Statistics, 1962.

Graham, N., Introduction to Computer Science, A Structured Approach, 1979.

Kaplan, A., Error Analysis of Hydrographic Positioning and the Application of Least Squares, M. S. Thesis, Naval Postgraduate School, 1980.

Mikhail, E. M., Observations and Least Squares, IEP, 1976.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, VA 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, CA 93940	2
3.	Chairman (Code 68Mr) Department of Oceanography Naval Postgraduate School Monterey, CA 93940	1
4.	Chairman (Code 63Rd) Department of Meteorology Naval Postgraduate School Monterey, CA 93940	1
5.	LCDR Dudley Leath (Code 68Lf) Department of Oceanography Naval Postgraduate School Monterey, CA 93940	1
6.	LCDR Francisco Silva Instituto Hidrografico Rua Das Trinas, 49 Lisboa Portugal	2
7.	Director Naval Oceanography Division Naval Observatory 34th and Massachusetts Avenue NW Washington, D.C. 20390	1
8.	Commander Naval Oceanography Command NSTL Station Bay St. Louis, MS 39522	1
9.	Commanding Officer Naval Oceanographic Office NSTL Station Bay St. Louis, MS 39522	1

No. Copies

10.	Commanding Officer Naval Ocean Research and Development Activity NSTL Station Bay St. Louis, MS 39522	1
11.	Chairman, Oceanography Department U.S. Naval Academy Annapolis, MD 21402	1
12.	Chief of Naval Research 800 N. Quincy Street Arlington, VA 22217	1
13.	Office of Naval Research (Code 480) Naval Ocean Research and Development Activity NSTL Station Bay St. Louis, MS 39522	1
14.	Director (Code PPH) Defense Mapping Agency Bldg. 56, U.S. Naval Observatory Washington, D.C. 20305	1
15.	Director (Code HO) Defense Mapping Agency Hydrographic Topographic Center 6500 Brookes Lane Washington, D.C. 20315	1
16.	Director (Code TSD-MC) Defense Mapping School Ft. Belvoir, VA 22060	1
17.	Director National Ocean Survey (OA/C) National Oceanic and Atmospheric Administration Rockville, MD 20852	1
18.	Chief, Program Planning and Liaison (NC2) National Oceanic and Atmospheric Administration Rockville, MD 20852	1

		No. Copies
19.	Associate Director, Marine Surveys and Maps (OA/C3) National Oceanic and Atmospheric Administration Rockville, MD 20852	1
20.	Chief, Hydrographic Surveys Division (OA/C35) National Oceanic and Atmospheric Administration Rockville, MD 20852	1
21.	Director Pacific Marine Center - NOAA 1801 Fairview Avenue East Seattle, WA 98102	1
22.	Director Atlantic Marine Center - NOAA 439 W. York Street Norfolk, VA 23510	1
23.	Commanding Officer Oceanographic Unit One USNS BOWDITCH (T-AGS21) Fleet Post Office New York, NY 09501	1
24.	Commanding Officer Oceanographic Unit Two USNS DUTTON (T-AG22) Fleet Post Office San Francisco, CA 96601	1
25.	Commanding Officer Oceanographic Unit Three USNS H. H. HESS (T-AGS38) Fleet Post Office San Francisco, CA 96601	1
26.	Commanding Officer Oceanographic Unit Four USNS CHAUVENT (T-AGS29) Fleet Post Office San Francisco, CA 96601	1
27.	Commanding Officer Mobility Logistic Division Marine Corps Development and Education Command Quantaco, VA 22134	1

No. Copies

28.	Commanding Officer H & S Company 8th Engineering Support Battalion Camp Le June, N.C. 29542	1
29.	Commanding Officer Oceanographic Unit Five USS HARKNESS Fleet Post Office New York, NY 09573	1
30.	IHO/FIG International Advisory Board International Hydrographic Bureau Avenue President J. F. Kennedy Monte-Carlo, Monaco	1
31.	Hydrographer of the Navy Ministry of Defense Hydrographic Department Taunton, Somerset TA129N England	1
32.	Hydrographer, Royal Australian Navy Hydrographic Office Box 1332 North Sydney 2060 New South Wales -Australia	1
33.	Dominion Hydrographer Canadian Hydrographic Service 615 Booth Street Ottawa, Ontario Canada	1
34.	Kapala Jawatan Hidro Oseanografi Jalan Gunuang Sahari-87 Jakarta, Indonesia	1
35.	Direccion General De Oceanografia Departmento De Hidragraficia Ave COYOACAN 131 Mexico 12, D.F.	1

No. Copies

36. Director 2
Instituto Hidrografico
Rua Das Trinas, 49
Lisboa
Portugal

37. Director De Hidrografia y Navegacion de la Marina 1
Direccion de Hidrografia y Navegacion de la Marina
Calle Saenz Pina -5ta Cuadra-La Punta
Callao-5 Peru

38. LCDR Gerald Mills (Code 68Mi) 1
Naval Postgraduate School
Monterey, CA 93940

39. Civil Engineering Department 1
Ports and Lighthouses Administration
Ras Eltin-Gate 1
Alexandria, Egypt

40. Hydrographic Service 1
Athens BST 902
Greece

41. ENG Antionio C. Silva 1
RUA Fialho De Almeida, 16 - 2
1000 - Lisboa
Portugal

42. Professor E. B. Thornton (Code 68Tm) 1
Department of Oceanography
Naval Postgraduate School
Monterey, California 93940

43. Antonio Ruiz Canavate 1
Teniente De Navio
Instituto Hidrografico De La Marina
Cadiz
Spain

44. Teniente Hector Soldi 1
Direccion De Hidrografia Y Navigacion De La Marina
Esquina Roca Y Gamarra
Chucuito - Callao
Peru

	No. Copies
45. Michael J. Ellet DMAHTC/Satellite Geodesy Attn: GSGS 6500 Brooks Lane Washington, D.C. 20315	1